

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



DESENVOLVIMENTO DE UMA SOLUÇÃO DE BUSINESS INTELLIGENCE

Mestrado em Engenharia Informática
Especialização em Sistemas de Informação

Pedro Miguel Martins Assis

Trabalho de projeto orientado por:
Prof. Doutor Luís Manuel Ferreira Fernandes Moniz e Eng. Rui Martinho

2017

Agradecimentos

Este trabalho representa o alcançar de mais um objetivo e como em todos os outros, não foi alcançado sozinho. Assim, quero deixar o meu agradecimento a algumas das pessoas que contribuíram de uma forma mais direta para o alcançar do mesmo.

Em primeiro lugar quero agradecer aos meus pais por todo o apoio que me deram ao longo dos anos e me incentivaram sempre para alcançar todos os meus objetivos e lutar sempre pelo melhor.

Quero também agradecer a toda a minha família que sempre me apoiaram e encorajaram em todos os momentos deste trabalho.

Um obrigado também para ao professor Luís Moniz por toda a disponibilidade e apoio demonstrados ao longo deste trabalho.

À Unipartner um grande obrigado por me proporcionarem uma experiência muito enriquecedora tanto a nível profissional como pessoal. De todos os meus colegas quero ainda deixar um agradecimento especial ao Rui Martinho, Hugo Figueira e Rosário Crujo que foram aqueles que mais me acompanharam no decorrer deste trabalho, ensinando-me e desafiando-me todos os dias.

À minha família.

Resumo

Este projeto, realizado no âmbito da Dissertação/Projeto de Engenharia Informática (DPEI), tem como objetivo desenvolver uma solução de *Business Intelligence* (BI).

Uma solução de *Business Intelligence* consiste num conjunto de estratégias e processos que são utilizados para recolher, tratar, analisar e apresentar informação relevante para a tomada de decisões. É construída utilizando ferramentas e sistemas que desempenham um papel importante no processo de planeamento estratégico de uma organização.

Estas ferramentas tecnológicas, que consistem num conjunto de aplicações de apoio à tomada de decisão, refletem um conjunto de estratégias e processos que, servindo-se dos dados disponíveis, permitem facilitar a análise e divulgação de informações de negócio, transformando grandes quantidades de informação em conhecimento útil, para ajudar executivos e gestores a tomarem decisões de negócio mais informados. Devem também garantir que a informação está sempre disponível e é apresentada de forma consistente.

O projeto surgiu na sequência de um convite realizado à Unipartner por um cliente do ramo da restauração que, face a uma consolidação tecnológica do grupo, obteve acesso a um volume considerável de informação relevante para o negócio. Face ao desafio de incluir estes novos dados nos seus modelos de *reporting* e tomada de decisão, reconheceu a necessidade de criar novos processos de normalização, tratamento e análise, que permitam a obtenção mais completa de informação sobre clientes e de negócio que suportem a tomada de decisões.

A solução a implementada suporta diversas análises de negócio efetuadas sobre as áreas de compras, fidelização, vendas, contabilidade, recursos humanos e qualidade. Sendo que é importante que forneça uma visão consolidada e estruturada dos dados residentes em diversas fontes de dados que o cliente possui.

No final do projeto esta solução é o ponto central e de confiança para que o cliente execute todas as suas análises ao negócio.

Palavras-chave: *Business Intelligence*; *data warehouse*; informação; *reporting*; tomada de decisões

Abstract

This project, carried out within the scope of the Dissertação/Projeto de Engenharia Informática (DPEI), aims to create a Business Intelligence solution.

A Business Intelligence solution consists of a set of strategies and processes used to collect, process, analyze and present information relevant to decision making. It is made up of tools and systems that play an important role in the strategic planning process of an organization.

These technological tools, which consist of a set of applications to support decision-making, reflect a set of strategies and processes that, using the available data, facilitate the analysis and dissemination of business information, transforming large amounts of information into Knowledge, helping executives and managers to make more informed business decisions. They must also ensure that information is always available and is consistently presented.

The project was created as a result of an invitation made to Unipartner by a customer in the restaurant industry, which, faced with a technological consolidation of the group, gained access to a considerable amount of information relevant to the business. Challenged to include these new data in its reporting and decision-making models, it was recognized the need to create new standardization, treatment and analysis processes that allow more complete information on customers and business to support decision making.

The solution to be implemented must support several business analyzes carried out in the areas of purchasing, loyalty, sales, accounting, human resources and quality. It is important that the final solution provides a consolidated and structured view of the data residing in various data sources that the customer has.

At the end of the project, the solution is expected to be the central and trustworthy source for the customer to perform all of their business analysis.

Keywords: Business Intelligence; data warehouse; information; reporting; decision-making

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Motivação	1
1.2	Objetivos.....	1
1.3	Instituição de acolhimento.....	2
1.4	Contribuições.....	2
1.5	Estrutura do documento.....	3
Capítulo 2	Conceitos.....	5
2.1	Solução de <i>Business Intelligence</i>	5
2.2	Fontes de dados operacionais	6
2.3	Processos de ETL	6
2.4	Área de apresentação de dados (<i>Data warehouse</i>).....	6
2.5	Aplicações de BI.....	7
2.6	<i>Data marts</i>	7
2.7	Modelo Multidimensional	8
Capítulo 3	O Trabalho.....	11
3.1	Ambiente de trabalho.....	11
3.1.1	Metodologia de trabalho.....	11
3.1.2	Tecnologias utilizadas	12
3.2	Arquitetura da solução.....	13
3.2.1	Cenários de implementação	14
3.2.2	Abordagem selecionada	17
3.3	Identificação de Requisitos.....	18
3.3.1	Modelo de Indicadores	18
3.3.2	Modelo de Qualidade	20
3.4	Processos de ETL	23
3.4.1	Processos de Extração (E).....	25
3.4.2	Processos de Transformação e Carregamento (TL)	35

3.5	Base de Dados Multidimensional	46
3.5.1	Modelo de Indicadores	47
3.5.1	Modelo de Qualidade	59
3.6	Processo de execução diário	68
3.7	Camada de Visualização.....	70
Capítulo 4	Conclusão e trabalho futuro	73
4.1	Principais Contribuições.....	73
4.2	Competências Adquiridas.....	74
4.3	Principais Dificuldades.....	74
4.4	Trabalho Futuro	75
Bibliografia	77
Apêndice A	79
4.5	Master Package.....	79

Lista de Figuras

Figura 1 - Estrutura típica de uma solução de Business Intelligence.....	5
Figura 2 - Data marts na arquitetura da solução	7
Figura 3 - Esquema em estrela.....	9
Figura 4 - Estrutura de um cubo de dados	9
Figura 5 - Processo de trabalho utilizando uma metodologia Ágil.....	12
Figura 6 - Arquitetura da solução	14
Figura 7 - Arquitetura do projeto com divisão entre on-premises e cloud	17
Figura 8 - Representação esquemática das dimensões envolvidas no modelo de indicadores.....	20
Figura 9 - Representação do modelo de qualidade - avaliação da distribuição	21
Figura 10 - Representação do modelo de qualidade - avaliação dos produtos	22
Figura 11 - Representação das bases de dados existentes na solução e suas relações	24
Figura 12 - Pacote de extração exemplo para identificar os seus componentes	25
Figura 13 - Estrutura do processo de extração de dados para a criação de dimensões	26
Figura 14 - Janela de configuração do EST - StoreProcedure	27
Figura 15 - Componentes envolvidos no passo de extração dos dados	28
Figura 16 - Representação da estrutura do pacote de extração dos dados que originarão as tabelas de factos	29
Figura 17 - Vista das variáveis criadas no pacote de Integration Services	30
Figura 18 - Configuração do componente EST - SQL Date Filter Table	32
Figura 19 - Configuração do componente DFT – Data Source to ODS	33
Figura 20 - Representação da estrutura do pacote SSIS_E_WebService_ExchangeRate	34
Figura 21 - Representação das variáveis existentes no SSIS_E_WebService_ExchangeRate	35
Figura 22 - Representação do processo de TL das dimensões.....	36
Figura 23 - Estrutura do comando Merge utilizado na inserção e atualização dos dados em EDW.....	38

Figura 24 - Representação do processo de TL dos factos de qualidade – pacote SSIS_TL_QUALITY_FactQualityAvDistribution	39
Figura 25 - Estrutura do processo T do modelo de indicadores.....	43
Figura 26 - Estrutura do processo TL do modelo de indicadores	45
Figura 27 - Modelo de indicadores com o requisito de segurança implementado..	47
Figura 28 - Configuração da Role utilizada no modelo de indicadores.....	48
Figura 29 - Modelo implementado na construção do cubo dos indicadores.....	49
Figura 30 - Representação das hierarquias da dimensão data.....	51
Figura 31 - Hierarquia da dimensão indicador.....	54
Figura 32 - Código MDX utilizado no cálculo dos indicadores complexos.....	57
Figura 33 - Código MDX para o cálculo do período homólogo, percentagem entre o valor realizado e o orçamentado, percentagem entre o valor realizado e o homólogo, diferença entre o valor realizado e o orçamentado e a diferença entre o valor realizado e o homólogo	58
Figura 34 - Código MDX utilizado no cálculo do número de lojas, média e ranking	58
Figura 35 - Esquema do modelo de qualidade incluindo os dois tipos de avaliação	60
Figura 36 - Modelo utilizado na construção do cubo de qualidade	61
Figura 37 - Código MDX utilizado no cálculo das métricas referentes à avaliação da distribuição.	68
Figura 38 - Estrutura do MasterPackage da solução.....	69
Figura 39 - Estrutura do MasterPackageExecution.....	70
Figura 40 - Exemplo 1 de um dashboard	71
Figura 41 - Exemplo 2 de um dashboard	72
Figura 42 - Exemplo 3 de um dashboard	72

Lista de Tabelas

Tabela 1 - Exemplo de uma matriz de processos.....	8
Tabela 2 - Estrutura da tabela DataSourceExtractionConfiguration da base de dados de configuração.....	29
Tabela 3 - Exemplo do comportamento de uma dimensão do tipo 2 com a ocorrência de um update.....	38
Tabela 4 - Dimensão data.....	50
Tabela 5 - Dimensão Loja.....	52
Tabela 6 - Dimensão Indicador.....	53
Tabela 7 - Dimensão Utilizador.....	54
Tabela 8 - Dimensão semana operacional.....	55
Tabela 9 - Tabela ponte entre a dimensão Utilizador e a dimensão Loja.....	55
Tabela 10 - Tabela ponte entre a dimensão Utilizador e a dimensão Indicador.....	56
Tabela 11 - Tabela de factos do modelo de indicadores.....	57
Tabela 12 - Dimensão motivo.....	62
Tabela 13 – Dimensão número de remessa.....	62
Tabela 14 - Dimensão distribuidor.....	63
Tabela 15 – Dimensão produto.....	63
Tabela 16 - Dimensão matricula.....	64
Tabela 17 - Dimensão parâmetro.....	64
Tabela 18 - Dimensão avaliação_distribuição.....	65
Tabela 19 - Tabela ponte AvaliaçãoDistribuição_Motivo.....	65
Tabela 20 - Tabela de Factos Avaliação Distribuição.....	66
Tabela 21 - Tabela de Factos Avaliação Produtos.....	67

Lista de acrónimos

Acrónimo	Significado
BI	<i>Business Intelligence</i>
BT	<i>Bridge Table</i>
CAPEX	<i>Capital expenditure</i>
DPEI	Dissertação/Projeto de Engenharia Informática
EDW	<i>Enterprise data warehouse</i>
EMEA	<i>Europe, the Middle East and Africa</i>
ERP	<i>Enterprise resource planning</i>
ETL	<i>Extraction-Transformation-Loading</i>
KPI	<i>Key Performance Indicator</i>
MDX	<i>Multidimensional Expressions</i>
ODS	<i>Operational data store</i>
OLAP	<i>OnLine Analytical Processing</i>
OLTP	<i>OnLine Transaction Processing</i>
OPEX	<i>Operational expenditure</i>
PEI	Projeto de Engenharia Informática
PK	<i>Primary Key</i>
SGBD	Sistema de Gestão de Base de Dados
SQL	<i>Structured Query Language</i>
SSAS	<i>SQL Server Analysis Services</i>
SSIS	<i>SQL Server Integration Service</i>
T	<i>Transformation</i>
TL	<i>Transformation-Loading</i>

Capítulo 1

Introdução

1.1 Motivação

O projeto de desenvolvimento de uma solução de *Business Intelligence* surgiu na sequência de um convite realizado à Unipartner por um cliente da área da restauração.

Atualmente o cliente possui apenas um conjunto de processos semiautomáticos e não integrados para suportar as diversas análises de negócio, sendo que estes processos apenas permitem obter visões parciais baseadas na informação existente nos diferentes sistemas operacionais.

A introdução de uma aplicação móvel direcionada especificamente para os clientes, promovendo a sua fidelização e premiando a sua frequência, fez aumentar significativamente a quantidade de informação sobre os clientes, incluindo os locais habituais onde tomam as suas refeições e as suas escolhas.

Tendo em conta o surgimento de novos requisitos de *reporting* para suporte do negócio, existe a necessidade de construir uma solução que permita a recolha, validação, categorização e armazenamento dos dados provenientes de todas as fontes de dados da organização, bem como a disponibilização dos mesmos para as novas análises feitas pelos utilizadores.

1.2 Objetivos

Construir uma solução de *Business Intelligence* que permita ao cliente obter uma visão da informação de Negócio, consolidada e devidamente estruturada, para facilitar o processo de análise estratégica e tomada de decisão. Para tal é necessário consolidar e estruturar a informação residente nos diversos sistemas operacionais, bem como permitir ao cliente ter acesso a um conjunto de processos automáticos que criem relatórios para suportar as suas análises de negócio e ainda processos de exploração manual.

É pretendido que as áreas de negócio abrangidas por esta implementação sejam várias, todas de grande importância para o sucesso do cliente, mas que endereçam

vertentes de negócio distintas. Essas áreas são: Compras, Fidelização, Vendas, Contabilidade, Recursos Humanos e Qualidade, para as quais têm de ser criados modelos detalhados e um modelo de indicadores que permitem análises específicas.

1.3 Instituição de acolhimento

O projeto foi realizado na Unipartner IT Services, que é uma empresa de consultoria informática que presta serviços ao setor público, tanto à administração central como à administração local, às empresas públicas, e ao sector privado, mais concretamente a empresas dos ramos: financeiro, indústria, comércio e serviços.

Os serviços prestados aos setores referidos assentam em quatro áreas de Negócio: consultoria em transformação digital, soluções de *cloud* e infraestruturas, desenvolvimento aplicacional e *outsourcing*. É de referir que a Unipartner é parceira de referência da Microsoft, possuindo o estatuto mais elevado de certificação – Gold –, e assim, sempre que possível, os serviços prestados utilizam predominantemente tecnologia deste fabricante.

A Unipartner é uma empresa que surgiu como consequência da reorganização da operação da multinacional Unisys na zona EMEA que ocorreu no ano de 2015, e que resultou na cessação da presença direta em Portugal.

É formada por cerca de cem colaboradores, predominantemente ex-colaboradores Unisys, e mantém-se como o parceiro de serviços Unisys para Portugal, de forma a garantir a continuidade dos projetos e a manutenção das relações com os diversos clientes e conserva uma visão e cultura similares.

1.4 Contribuições

O trabalho realizado no âmbito do meu Projeto de Engenharia Informática (PEI) consistiu na participação da conceção de uma solução de *Business Intelligence*, criada para um cliente da empresa Unipartner, com o objetivo de fornecer informação de negócio e uma visão consolidada respeitante a todas as áreas do seu negócio.

O meu contributo focou-se na construção dos modelos de qualidade e de indicadores, sendo estes os modelos abordados neste documento. No entanto, também estive envolvido ativamente na construção de todos os outros modelos e elementos que constituem a solução, tais como processos de ETL e estruturas necessárias para os mesmos, a criação e estruturação da base de dados multidimensional e, finalmente, a construção de *dashboards*

1.5 Estrutura do documento

Este documento está dividido em quatro capítulos organizados da seguinte forma:

- Capítulo 1 – É efetuada toda a contextualização deste trabalho através da sua motivação e objetivos, para além da caracterização da instituição de acolhimento onde o projeto foi desenvolvido.
- Capítulo 2 – São explicados os conceitos teóricos base relacionados com a construção de uma solução de BI.
- Capítulo 3 – Descreve de forma detalhada todo o trabalho realizado, dando destaque ao ambiente de trabalho, tecnologias utilizadas, arquitetura da solução, identificação dos vários componentes que constituem a solução (pacotes, processos, tarefas, tabelas, etc), base de dados multidimensional respetivos *dashboards*.
- Capítulo 4 – Enumera as principais contribuições que a solução proporcionou ao cliente, as competências adquiridas ao longo de todo o projeto, as principais dificuldades enfrentadas e por fim o trabalho futuro.

Capítulo 2

Conceitos

2.1 Solução de *Business Intelligence*

Uma solução de *Business Intelligence* consiste num conjunto de estratégias e processos que são utilizados para recolher, tratar, analisar e apresentar informação relevante para a tomada de decisões de uma organização. Esta deve garantir que a informação está sempre acessível e é apresentada de forma consistente.

Uma solução de *Business Intelligence* é, normalmente, composta por quatro componentes distintos:

- Fontes de dados operacionais;
- Sistema/estrutura de Extração, Transformação e Carregamento ou *Loading* (ETL);
- Área de apresentação de dados;
- Aplicações de BI.

A Figura 1 permite observar como é que estes quatro componentes se interligam entre si para criar a estrutura típica de uma solução de *Business Intelligence*.

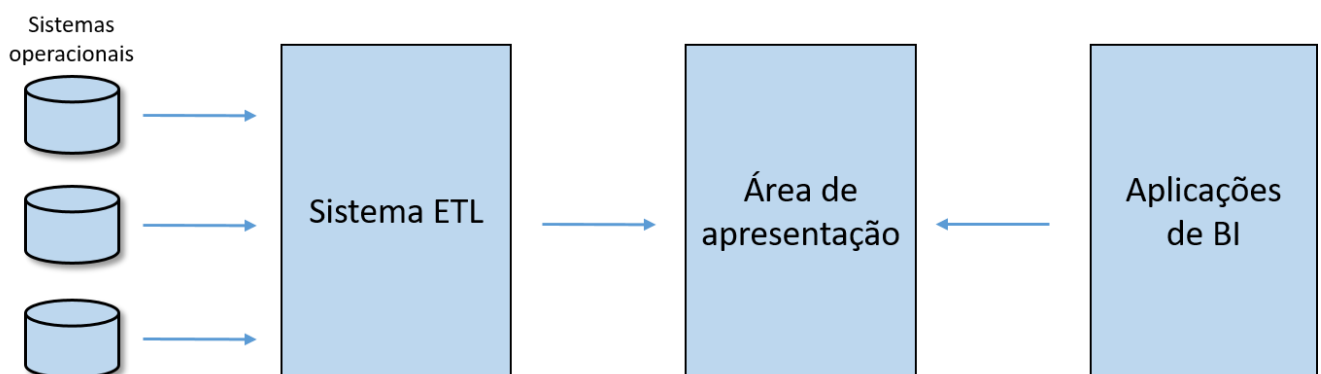


Figura 1 - Estrutura típica de uma solução de *Business Intelligence*

2.2 Fontes de dados operacionais

Os sistemas operacionais, tais como OLTP e ERP, são os responsáveis por fazerem os dados entrar na organização, podendo estes ser fontes internas ou externas. Estes sistemas são otimizados para processos repetitivos de escrita e leituras simples de dados, tendo por isso um modelo de dados complexo e normalizado.

Tendo em conta estas características, é facilmente perceptível que fazer interrogações analíticas sobre estes sistemas é uma atividade bastante pesada e exigente, devendo assim serem reduzidas ao máximo, pois a sua estrutura não é adequada a este tipo de operações.

2.3 Processos de ETL

Os processos de ETL são compostos por todos os componentes que existem entre os sistemas operacionais e a área de apresentação de dados. Esta estrutura tem três grandes etapas:

- **Extração** – é feita a cópia dos dados do sistema operacional, podendo esta ser guardada numa área de *staging* (área temporária sobre a qual são realizadas as transformações).
- **Transformação** – são realizados todo o tipo de operações necessárias sobre os dados extraídos até que estes atinjam a forma esperada. Estas operações podem ser, por exemplo, limpeza e/ou uniformização de campos, lidar com elementos em falta, entre outros.
- **Carregamento (*Loading*)** – os dados são carregados para a área de apresentação tendo em conta a estrutura definida na área de apresentação.

2.4 Área de apresentação de dados (*Data warehouse*)

Na área de apresentação, os dados são organizados, armazenados e disponibilizados para consumo de toda a organização. Nesta área são utilizados índices e valores pré-calculados com o objetivo de reduzir o tempo de resposta aos pedidos recebidos.

O acesso aos dados pode ser feito através que interrogações diretas, escrita de relatórios ou ainda aplicações de BI.

Esta área é a única que permite aos utilizadores acederem aos dados, o que quer dizer que todos os processos anteriores são invisíveis para o utilizador.

2.5 Aplicações de BI

Qualquer aplicação que faça interrogações à área de apresentação do *data warehouse* é considerada uma aplicação de BI [11].

Uma aplicação de BI pode ser algo bastante simples como uma interrogação ou pode ser tão complexo como alguns modelos de exploração de dados (*data mining*).

Estas aplicações, muitas vezes proporcionam aos utilizadores ferramentas para tornarem a leitura dos dados bastante fácil e intuitiva através da construção dos mais variados tipos de gráficos.

2.6 Data marts

Na construção de um *data warehouse* existe outro conceito bastante importante, os *data marts*. Estes são um subconjunto/vistas do *data warehouse* e normalmente são orientados a uma área específica do negócio da organização, por exemplo, organização departamental. Por serem mais pequenos e focados são, por isso, mais eficientes.

Os *data marts* permitem também controlar o acesso aos dados, pois é possível definir que apenas um determinado utilizador ou grupo de utilizadores lhes consegue aceder.

A Figura 2 mostra em que local da arquitetura da solução é que os *data marts* se encontram.

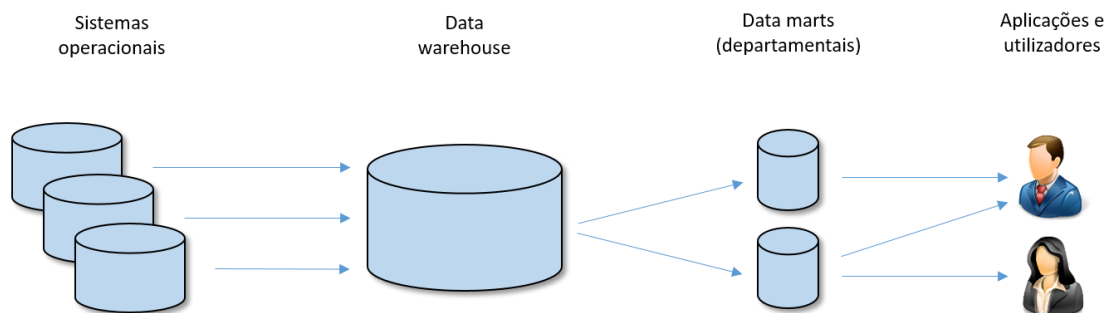


Figura 2 - Data marts na arquitetura da solução

Para o desenho dos *data marts* é, frequentemente, utilizada uma matriz de processos [9].

Nesta matriz cada linha representa um processo do negócio e cada coluna representa as dimensões envolvidas na análise de cada processo. Um *data mart* corresponde, normalmente a uma linha da matriz [14].

A Tabela 1 é um exemplo de uma matriz de processos.

Processo de negócio \ Dimensões	Data	Empresa	Cliente	Produto	Localização	Fornecedor
Compras de clientes	X	X	X	X	X	
Encomendas	X	X		X		X
Custo Produto	X	X	X	X	X	X
Desconto a clientes	X		X	X	X	
Inventário	X			X	X	

Tabela 1 - Exemplo de uma matriz de processos

É possível observar que existem algumas dimensões que são utilizadas em vários *data marts*. Nestes casos é necessário que a dimensão seja uma dimensão conformada. Uma dimensão diz-se conformada quando duas cópias da dimensão são exatamente iguais ou uma é um subconjunto perfeito da outra. Para a criação destas dimensões é necessário que todos os departamentos da organização estejam de acordo na sua definição e no que cada uma representa.

2.7 Modelo Multidimensional

Os modelos multidimensionais são utilizados com o objetivo de construir uma estrutura de dados de fácil utilização e compreensão para os utilizadores, sendo também otimizados para dar resposta às interrogações [15]. É normal existir pelo menos um modelo multidimensional por cada *data mart* existente.

Estes modelos são, normalmente, modulados num esquema em estrela dando posteriormente origem a cubos de dados.

O esquema em estrela é implementado em bases de dados relacionais e o nome vem da forma que as relações das tabelas aparentam. O modelo é composto por várias tabelas de dimensão ligadas a uma tabela de factos ao centro, tal como é esquematizado na Figura 3.

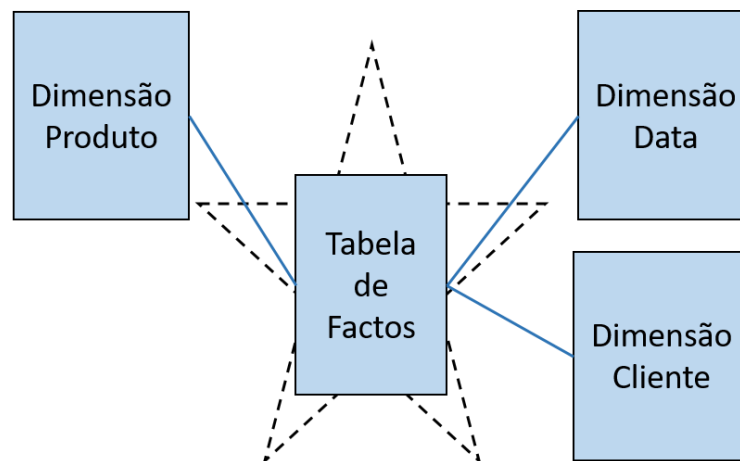


Figura 3 - Esquema em estrela

Os cubos de dados (Figura 4) são estruturas implementadas em bases de dados multidimensionais que permitem analisar os dados de diferentes perspetivas e detalhes. Isto deve-se ao facto de ser possível analisar os factos disponíveis pelas diferentes dimensões consideradas. É ainda possível realizar várias operações sobre o cubo com o fim de obter diversos níveis de detalhe sobre os dados existentes.

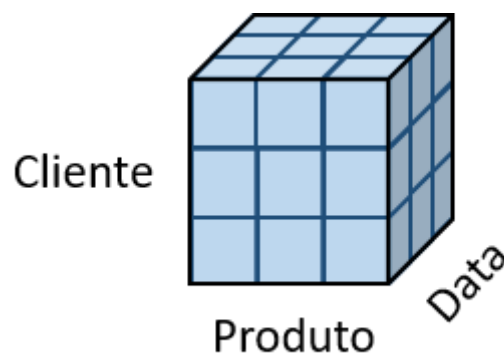


Figura 4 - Estrutura de um cubo de dados

No desenvolvimento de uma solução de *Business Intelligence* é usual que a construção do esquema em estrela seja o primeiro passo da modelação multidimensional e a criação/população de um cubo com os dados provenientes desse esquema seja o passo seguinte.

Capítulo 3

O Trabalho

3.1 Ambiente de trabalho

3.1.1 Metodologia de trabalho

Com o objetivo de desenvolver o projeto num curto espaço de tempo e com níveis de qualidade elevados, a equipa utilizou o *Scrum* como processo de desenvolvimento assente numa metodologia Ágil [16], garantindo a agilidade e flexibilidade necessárias à adaptação do projeto a alterações e a novos pedidos, tal como pretendido pelo cliente pois o seu Negócio está em constante evolução e a apresentar novas necessidades.

Com a utilização desta metodologia existem definidos três papéis importantes:

- **Product Owner** – é o representante do cliente para o qual a solução é desenvolvida, tendo a seu cargo a identificação dos requisitos funcionais e não funcionais pretendidos. Estes requisitos são colocados numa lista, normalmente denominada como *backlog*.
- **Scrum Master** – tem como função garantir um bom canal de comunicação com o *Product Owner* de forma a que todos os requisitos definidos sejam corretamente entendidos e ordenados, realizar a análise dos processos a implementar, garantir a correta interpretação dos mesmos pela Equipa e controlar o processo de desenvolvimento para garantir que os ciclos de desenvolvimento estão a ser cumpridos.
- **Equipa** – é o conjunto de pessoas responsável pela realização das tarefas de desenvolvimento e entrega da solução, respondendo a todos os requisitos identificados. A equipa apresenta características multifuncionais, de autoorganização e de autonomia, dividindo o conteúdo do projeto em ciclos de trabalho concentrados e curtos chamados *sprints*.

Neste projeto os *sprints* tiveram, normalmente a duração de uma a duas semanas, durante os quais foram implementados os requisitos que constam no *backlog*. O *backlog* foi sendo dividido em grupos de requisitos, designados como *sprint backlog*, para serem desenvolvidos em cada *sprint*.

No final de cada *sprint* foi realizada uma entrega e/ou efetuada uma demonstração da solução ao cliente, para que este tivesse a oportunidade de rever a sua lista de requisitos (*backlog*).

Nesta revisão o cliente pode proceder a alterações à lista de requisitos e às suas especificações, reordenando-a, efetuando a adição de novos requisitos, a alteração a requisitos existentes e/ou a remoção de requisitos.

No que respeita à organização da equipa, todos os dias foi realizada uma rápida reunião de ponto de situação e alinhamento, dando oportunidade a que todos os membros transmitissem o estado das suas tarefas e, caso necessário, fosse efetuado a sua redistribuição e replaneamento. O processo de trabalho descrito encontra-se ilustrado na Figura 5.

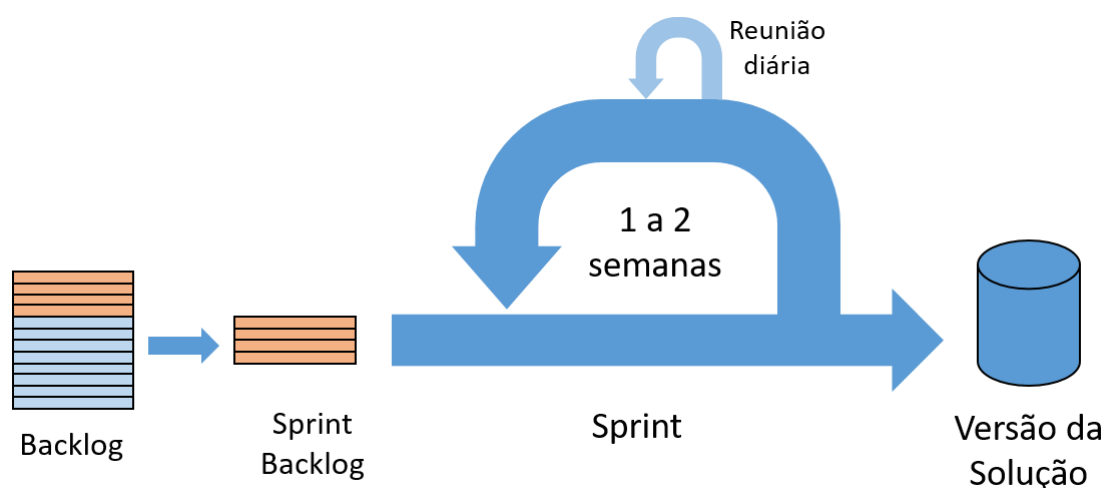


Figura 5 - Processo de trabalho utilizando uma metodologia Ágil

3.1.2 Tecnologias utilizadas

Nesta secção são descritas as tecnologias utilizadas no desenvolvimento da solução. Tal como referido anteriormente a Unipartner é parceira da Microsoft e, por esse motivo, todas as tecnologias utilizadas no decorrer do estágio e no desenvolvimento da solução de BI são tecnologias Microsoft.

SQL Server Database

O Microsoft SQL Server Database é um sistema de gestão de base de dados (SGBD) que tem como principal função armazenar, processar e manter os dados seguros [3][4]. No SQL Server Database é possível armazenar dados que façam parte de sistemas de Processamento de Transações em Tempo Real (OLTP), normalmente referidos como sistemas operacionais, bem como armazenar dados de sistemas que permitam realizar

uma *OnLine Analytical Processing* (OLAP), usualmente são utilizadas estruturas como o cubo de dados.

SQL Server Integration Services

É uma plataforma bastante utilizada para realizar extrações, transformações e carregamento de dados (ETL). O Integration Services tem um grande conjunto de transformações pré-definidos e é possível criar diversas soluções graficamente, sem necessidade de programar [5][6]. Caso as transformações existentes não sejam suficientes é possível criar novas transformações programando as ações necessárias.

SQL Server Analysis Services

O SQL Server Analysis Services (SSAS) é um motor de dados que fornece dados analíticos (OLAP) e ferramentas de exploração de dados (*data mining*) para aplicações de BI [7][8]. O Analysis Services suporta OLAP por permitir criar e gerir modelos multidimensionais. É possível também que aplicações como o Power BI consumam os dados existentes nestes modelos multidimensionais e criem os mais variados relatórios.

Power BI

Power BI é uma aplicação baseada na *cloud* que permite interagir com diversas fontes de dados e transformá-los nos mais variados gráficos e indicadores [1]. Ao juntar os gráficos e indicadores é possível construir relatórios interativos. Estes relatórios são acessíveis a partir de qualquer tipo de dispositivo.

3.2 Arquitetura da solução

A solução implementada foi concebida tendo em consideração a dispersão dos dados por dez sistemas independentes, com origens diversas e com características tecnológicas diferentes. Estes sistemas, que têm propósitos completamente distintos, constituem cada um deles ferramentas imprescindíveis para as áreas de negócio que servem. A arquitetura construída para acomodar a solução final está representada na Figura 6.

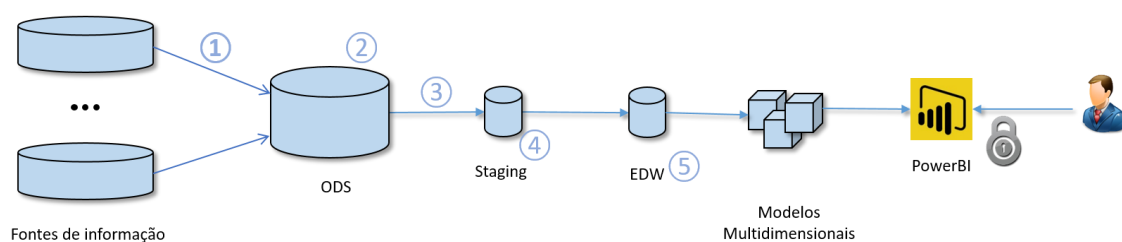


Figura 6 - Arquitetura da solução

O *data warehouse* é constituído por cinco módulos:

1. Camada de integração, onde é efetuada a integração com as dez fontes de informação identificadas
2. *Operational Data Store* (ODS) que irá possuir uma cópia sincronizada dos dados das diversas fontes de informação
3. Camada de transformação entre o ODS e *Staging*
4. Carregamento entre *Staging* e a base de dados EDW
5. Modelos multidimensionais para compras, vendas, contabilidade, recursos humanos, qualidade, fidelização e modelo de indicadores.

Os modelos multidimensionais disponibilizarão informação que será acedida através de relatórios construídos em Power BI.

3.2.1 Cenários de implementação

Tendo em conta as características e a arquitetura da solução preconizada é possível identificar três abordagens distintas que podem ser seguidas na sua implementação:

1. Solução *on-premises*;
2. Solução na *cloud* (Microsoft Azure);
3. Solução híbrida, parte *on-premises* e parte na *cloud* (Microsoft Azure);

A escolha da abordagem a seguir depende bastante de organização para organização, quais são os objetivos atuais, a sua projeção para o futuro e o crescimento esperado. Atualmente as tecnologias utilizadas em qualquer um dos cenários são muito semelhantes. Como tal, a escolha incide maioritariamente sobre os custos, o tempo de implementação e a escalabilidade de cada uma das soluções. Para permitir uma escolha sensata e escolher a melhor das abordagens é necessário perceber o que é que cada um dos cenários envolve.

Solução *on-premises*

Uma solução *on-premises* é uma solução que é implementada nas instalações da organização.

Para implementar uma solução *on-premises* é necessário, de uma forma geral, ter em consideração vários aspetos que são claramente importantes para a corporização da mesma e que têm de ser contemplados, bem como os custos associados a cada um deles:

- Servidores e sistemas de virtualização;
- Licenciamento dos produtos de *software*;
- Suporte e manutenção dos vários produtos envolvidos;
- Infraestruturas de comunicações locais de *Datacenter*;
- Ocupação do espaço de *Datacenter*;
- Outras necessidades acessórias ao funcionamento dos vários componentes (energia elétrica, arrefecimento, etc.);
- Pessoas, conhecimentos e tempo para realização da instalação de todo o ambiente e a sua configuração de raiz, ou contratação de serviços externos para a realização destas tarefas;
- Existência de uma equipa especialista em administração de sistemas para gestão, monitorização e resolução de eventuais problemas;
- Sistemas complementares para salvaguarda (*backup*) da informação.

Neste tipo de abordagem é necessário ainda ter em conta que:

- A escalabilidade da solução é dispendiosa e/ou difícil por ser necessário a aquisição de novos equipamentos e toda a sua configuração.
- Morosidade e custos inerentes às atividades de reestruturação e/ou redimensionamento da solução;
- Para garantir que em caso de ocorrência de uma catástrofe para que os dados não sejam perdidos é necessário que estes estejam replicados noutra local geográfico. Assim a organização tem que construir toda a estrutura necessária em duplicado ou contratar algum serviço na *cloud* que sirva como *backup*.

Esta é uma abordagem que assenta num modelo financeiro de investimento ou despesa de capital (*Capital Expenditure* - CAPEX).

Solução na *cloud* (Microsoft Azure)

Uma solução na *cloud* é quando os recursos utilizados, tanto processamento como armazenamento ocorrem remotamente, em equipamentos da entidade que presta o serviço.

O Microsoft Azure é uma plataforma *cloud* que fornece um conjunto extremamente alargado de serviços. Esta plataforma dá assim a hipótese de desenvolver e gerir qualquer solução pretendida para resolver qualquer tipo de desafio a nível de arquitetura como a nível de escala [12][13]. Escala esta que pode ser de armazenamento, processamento ou de tráfego.

Para implementar uma solução na *cloud*, utilizando o Microsoft Azure é necessário:

- Criar uma conta no Azure;
- Utilizar as ferramentas disponibilizadas e/ou criar as máquinas virtuais necessárias (existindo um grande conjunto de máquinas virtuais já pré configuradas) e acrescentar apenas as configurações necessárias;

O Microsoft Azure garante ainda algumas funcionalidades extremamente úteis para uma organização:

- Os dados armazenados são automaticamente replicados de forma a garantir que em caso de catástrofe os dados não são perdidos.
- Serviços com alta disponibilidade, visto que é a própria Microsoft que faz a manutenção e recuperação de todos os sistemas de virtualização.
- Os serviços do Azure possuem a elasticidade necessária para poderem ser aumentados ou diminuídos, de forma quase instantânea, para fazer face às necessidades do momento, sendo apenas cobrados os recursos que são utilizados.

Esta é uma abordagem que assenta num modelo financeiro de despesa operacional (*Operating Expenses* - OPEX).

Solução híbrida - *on-premises* e *cloud* (Microsoft Azure)

O Microsoft Azure permite a construção de soluções híbridas, ou seja, parte da solução desenvolvida nas instalações da organização (*on-premises*) e parte no Azure (*cloud*) combinando as características de cada uma das abordagens. Isto é possível pois o Azure em quase todos os seus serviços tem pontos de ligação para serviços *on-premises*.

Esta é uma abordagem que combina o modelo financeiro CAPEX com o modelo financeiro de despesa operacional OPEX.

3.2.2 Abordagem selecionada

Após a análise, de uma forma geral, de ambas as abordagens é fácil concluir que uma solução totalmente assente no Microsoft Azure seria a mais indicada por apresentar um desenvolvimento potencialmente mais rápido, bem como uma flexibilidade extremamente elevada, sendo apenas cobrado o serviço correspondente aos recursos utilizados. Porém a abordagem implementada foi a híbrida. Esta escolha foi tomada tendo em conta as imposições realizadas pelo cliente.

Assim a solução tem toda a parte do *data warehouse* desenvolvida *on-premises* e apenas a camada de visualização, o Power BI, será desenvolvida na *cloud*. Esta distribuição está demonstrada na Figura 7.

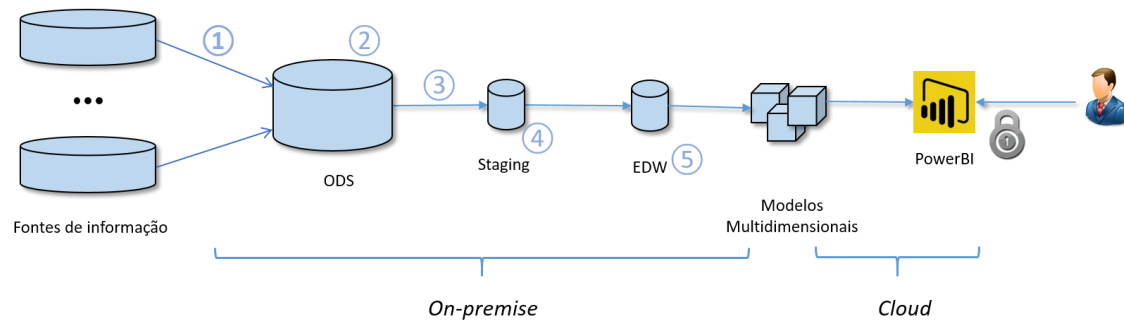


Figura 7 - Arquitetura do projeto com divisão entre on-premises e cloud

3.3 Identificação de Requisitos

Após ter sido tomada a decisão sobre qual a arquitetura da solução adotada, o passo seguinte, e ainda anterior ao início do desenvolvimento da solução, foi a identificação cuidadosa de todos os requisitos, tanto funcionais como não funcionais, dos modelos a implementar, tendo em vista a construção de um *data warehouse* corporativo, que permite a consolidação, estruturação e normalização da informação dispersa pelos diversos sistemas operacionais e possibilitando que posteriormente a informação seja acedida pelos utilizadores através de um *software* de visualização de informação.

Os requisitos funcionais são aqueles que retratam o comportamento esperado de cada um dos modelos, ou seja, tendo como base um determinado conjunto de dados é esperado que a solução apresente um determinado comportamento de forma a que seja possível atingir os resultados de análise pretendidos.

Os requisitos não funcionais, neste caso, são as exigências que o cliente apresenta relativamente ao funcionamento da solução e as suas necessidades.

Este trabalho de levantamento de requisitos foi desenvolvido em conjunto com o cliente de forma a que nenhuma funcionalidade ou regra de negócio fosse esquecida e pudesse ser refletida de acordo com o processo de negócio definido pelo cliente.

3.3.1 Modelo de Indicadores

O modelo de Indicadores tem como objetivo permitir a realização de uma análise transversal a todas as áreas da empresa tendo como nível base de análise e comparação a loja. As áreas cobertas por este modelo são a administrativa, operacional, vendas, contabilidade, recursos humanos e qualidade, sendo cada uma destas áreas analisada através de um conjunto definido de indicadores.

Como o cliente possui várias empresas e lojas espalhadas por diferentes países, tem a necessidade de ter mecanismos que permitam a realização de análises uniformes independentemente da localização das suas lojas. Para atingir este objetivo foi necessário criar dois mecanismos que tornem possível a comparação de lojas no mesmo período temporal de operação e ainda que permitam que a comparação seja efetuada utilizando a mesma moeda de referência. De uma forma mais clara, uma determinada semana do ano pode corresponder a semanas operacionais diferentes para cada uma das empresas, como tal, tem que existir um mecanismo para ser possível realizar análises por semana operacionais.

Estando o cliente presente em vários países, alguns deles com a moeda local diferente do Euro, foi necessário um mecanismo que permita a conversão do Euro para a moeda local de cada país onde existem lojas, e vice-versa.

O modelo serve como fonte de dados para a construção de diversos *dashboards*, os quais têm diferentes níveis de análise e públicos-alvo. Assim sendo, é necessário que exista uma camada de segurança. O objetivo desta camada é garantir que um determinado utilizador apenas tem acesso à informação para a qual tem permissões. Permissões estas que são definidas ao nível da loja e dos indicadores.

Outro dos objetivos deste modelo é o de permitir uma fácil comparação entre lojas para um mesmo indicador através de *rankings*. Para tal é necessário a implementação de diversos *rankings* de forma a permitir a análise da loja em diversos universos. Entende-se como universo de análise todas as lojas que tenham o mesmo valor para o campo sobre o qual o *ranking* é calculado. Foram implementados dez *rankings* distintos e totalmente dinâmicos, dos quais seis vêm com um contexto de análise pré-definido e os restantes quatro são definidos à posteriori pelo cliente. Os seis *rankings* pré-definidos são sobre os seguintes universos de análise: empresa, conceito, país, região, no supervisor e do supervisor. Por um *ranking* dinâmico entende-se que o cliente pode reajustar o universo de análise de cada um deles em qualquer altura.

Relativamente aos dados que dão origem aos indicadores de contabilidade, estes podem ser alterados até um ou dois meses após a data dos mesmos, assim é pretendido que a solução garanta uma janela temporal de quatro meses no cálculo dos mesmos de forma a garantir que todas as eventuais alterações foram consideradas.

Para além destes requisitos, foi acordada a implementação de inúmeras métricas sobre este modelo. A partir dos sistemas fonte do cliente é possível extrair o valor realizado e o valor orçamentado para cada um dos indicadores por loja e para cada data. Tendo estas duas métricas como base, foi acordada a implementação das seguintes métricas calculadas:

- Valor realizado no período homólogo
- Diferença entre o valor realizado e o valor orçamentado
- Diferença entre o valor realizado e período homólogo
- Percentagem do valor realizado face ao período homólogo
- Percentagem do valor realizado face ao valor orçamentado
- Média do valor realizado
- Número de lojas
- *Rankings* (vários universos de análise)
- *Rankings* com total (vários universos de análise)

Para concluir o levantamento de requisitos, de forma a garantir que estes foram bem identificados e compreendidos, foi construído um esquema com as dimensões envolvidas neste modelo, tal como é representado na Figura 8. Esta representação tem apenas como objetivo a apresentação das dimensões envolvidas e não a implementação do modelo no cubo.

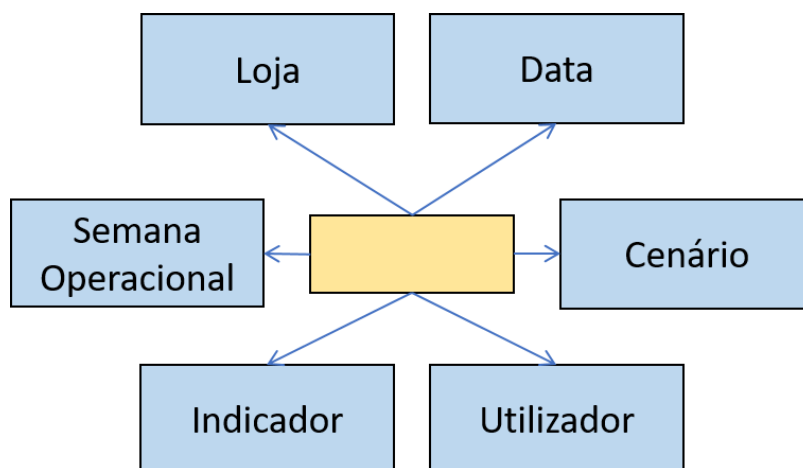


Figura 8 - Representação esquemática das dimensões envolvidas no modelo de indicadores

3.3.2 Modelo de Qualidade

O modelo de qualidade tem o objetivo de permitir a realização de análises sobre a qualidade das distribuições efetuadas pelos fornecedores e a qualidade dos produtos entregues nas lojas. Estas análises, da distribuição e dos produtos, são completamente distintas e independentes, pois são feitas após um sorteio realizado diariamente pelo sistema do cliente que atribui para cada loja o tipo de parâmetros e/ou produtos tem de ser conferidos na entrega.

Por análise da distribuição entende-se a avaliação das condições de transporte e de entrega, por exemplo, avaliando as condições da viatura (higiene e temperatura). Na análise dos produtos é avaliada a conformidade dos mesmos, por exemplo, as características físicas do mesmo, higiene e conservação.

Tendo em conta que a avaliação da distribuição e do produto são realizados de forma distinta e independente, o modelo de qualidade tem assim dois submodelos, um para cada uma das avaliações.

De uma forma mais detalhada a análise para o “modelo de qualidade – avaliação da distribuição” ocorre quando uma loja recebe de um distribuidor um conjunto de produtos numa determinada data. Sobre esta entrega é guardada a informação da distribuição, o número de remessa e a matrícula da viatura, e é feita uma avaliação de diversos

parâmetros (higiene, temperatura, ...) e cada um destes parâmetros é avaliado de acordo com a sua conformidade (conforme ou não conforme). No caso de serem avaliados como não conforme é necessário proceder à identificação do(s) motivo(s) a que levaram a tal avaliação.

Para identificar de forma mais clara as dimensões que são esperadas neste modelo foi criado o esquema apresentado na Figura 9 - Representação do modelo de qualidade - avaliação da distribuição.

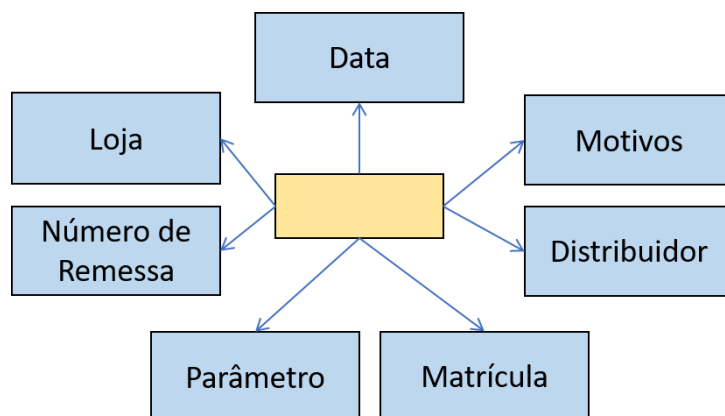


Figura 9 - Representação do modelo de qualidade - avaliação da distribuição

Ainda neste modelo de distribuição é esperada a implementação das seguintes métricas:

- Número de avaliações
- Número de conformidades
- Número de não conformidades
- Número de não verificações
- Número de devoluções
- Valor médio da temperatura
- Percentagem de avaliações conformes
- Percentagem de avaliações não conformes
- Percentagem de devoluções

No que diz respeito ao “modelo de qualidade – avaliação dos produtos” esta análise ocorre quando uma loja recebe um conjunto de produtos de um distribuidor numa determinada data. Desta entrega é guardada a informação de qual o número de remessa e o número do veículo que realizou a entrega. A cada produto entregue está associada a informação do seu produtor e fornecedor. Estes são avaliados de acordo com um conjunto de parâmetros e, quando algum destes parâmetros não está conforme, é necessário a

identificação do(s) respetivo(s) motivo(s) de tal não conformidade. É expectável que as avaliações sejam realizadas no momento da entrega dos produtos, contudo é possível que as mesmas sejam apenas realizadas posteriormente durante as atividades normais da loja.

O esquema observado na Figura 10 - Representação do modelo de qualidade - avaliação dos produtos tem como objetivo permitir uma fácil identificação de todas as dimensões que existem neste modelo.

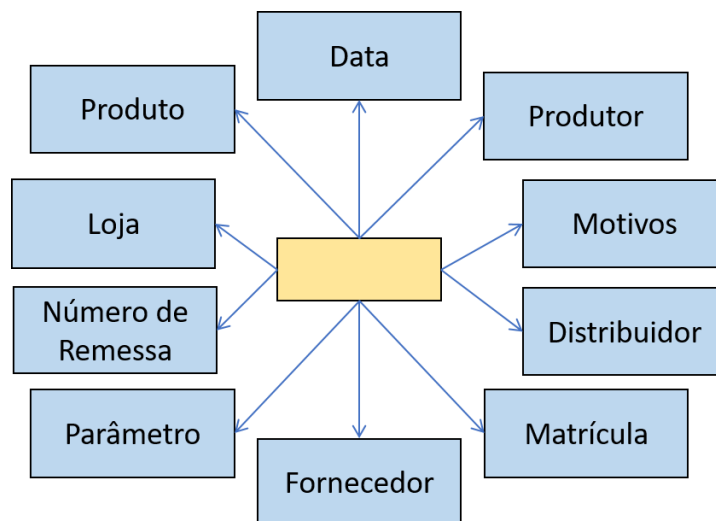


Figura 10 - Representação do modelo de qualidade - avaliação dos produtos

De forma a complementar a análise efetuada neste modelo serão ainda implementadas as seguintes métricas:

- Número de avaliações
- Número de conformidades
- Número de não conformidades
- Número de devoluções
- Valor médio da temperatura
- Percentagem de avaliações conformes
- Percentagem de avaliações não conformes
- Percentagem de avaliações não verificadas
- Percentagem de devoluções
- Quantidade recebida
- Quantidade comprada

3.4 Processos de ETL

Depois de concluída a identificação de requisitos teve início o desenvolvimento dos processos de extração, transformação e carregamento de dados para que estes sejam disponibilizados para a base de dados multidimensional. Estes processos foram criados utilizando *SQL Server Integration Service* (SSIS).

De forma a garantir uma boa organização e o devido isolamento de todos estes processos optou-se pela utilização de cinco bases de dados relacionais, em que cada uma delas é utilizada para uma fase específica do processo. Estas bases de dados e as suas respetivas tarefas são as seguintes:

- *Operational Data Store* (ODS) – utilizada no processo de extração, guardando uma cópia exata dos dados existentes nos sistemas operacionais e dando início aos processos de transformação nas suas *views*, nomeadamente a normalização e otimização do tipo de dados e tratamento de campos não preenchidos.
- *Staging* – realiza os restantes processos de transformação e efetua o pré-cálculo de algumas métricas com o objetivo de acelerar os processos de consulta subsequentes.
- *Enterprise Data Warehouse* (EDW) – contém os dados, já com a estrutura final, que são utilizados para alimentar a base de dados multidimensional.
- *Config* – usada para guardar as configurações a serem utilizadas pelos pacotes de *Integration Service* nos seus respetivos processos.
- *Logging* – utilizada para armazenar o registo de todas as operações feitas sobre as restantes bases de dados, essencialmente para questões de auditoria.

A Figura 11 representa uma esquematização das bases de dados existentes e as setas representam os fluxos existentes dos dados.

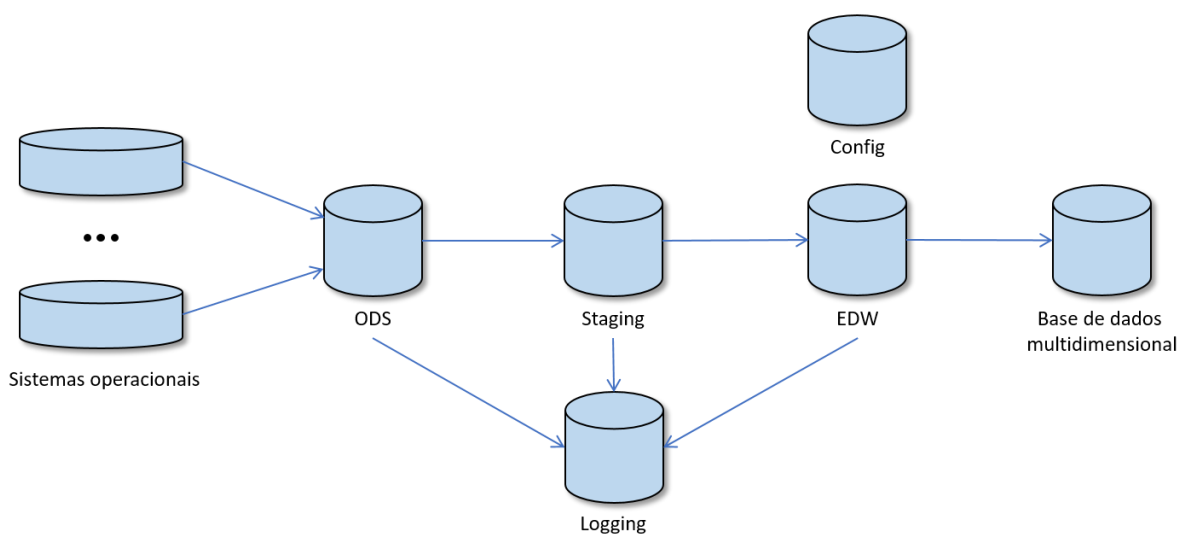


Figura 11 - Representação das bases de dados existentes na solução e suas relações

É de salientar que a base de dados EDW contém todos os dados existentes na solução e que nunca é apagada qualquer informação desta, enquanto que as bases de dados ODS e *Staging* são bases de dados de trabalho, ou seja, de cariz temporário onde sempre que o processo é executado os dados existentes nas mesmas são eliminados e são apenas tratados os novos dados.

A base de dados multidimensional, que já não faz parte do processo de ETL, é utilizada para manter as estruturas multidimensionais (cubos) de forma a permitir a realização das análises pretendidas.

Todos os processos de ETL criados necessitam de efetuar operações sobre as bases de dados da solução. Na realidade a solução trabalha sobre quinze bases de dados: as cinco anteriormente referidas e dez adicionais que dizem respeito a bases de dados do sistema operacional do cliente.

Para que cada pacote de *Integration Services* (IS) tenha acesso às bases de dados, foi necessária a criação de quinze *connection managers*. Estes são representações lógicas das ligações que o projeto de IS cria quando executa cada um dos seus pacotes. Os pacotes são constituídos por *containers* que, por sua vez, contêm as tarefas a executar para atingir um determinado propósito. Na Figura 12 está representada a estrutura de um pacote de extração onde é possível observar a relação entre os seus componentes.

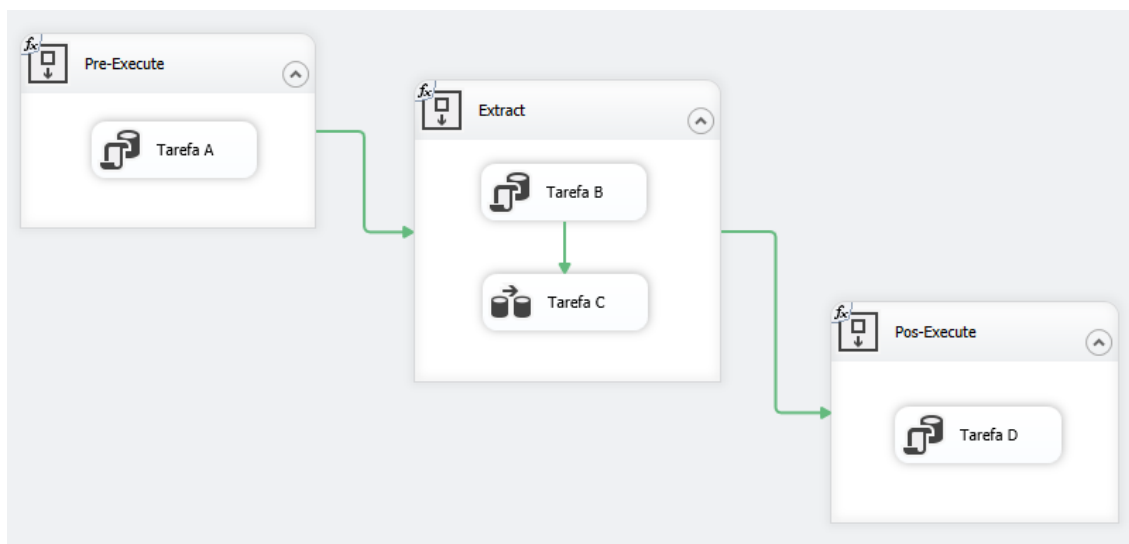


Figura 12 - Pacote de extração exemplo para identificar os seus componentes

Como melhor prática instituída na empresa, para estruturar e uniformizar os processos de ETL, prevendo a possibilidade de ser necessária a execução de tarefas que antecedem ou que precedem as tarefas de extração, transformação e carregamento, em todos os processos foram criados os *containers Pre-Execute* e *Pos-Execute*. No entanto, neste projeto em nenhum dos casos estes *containers* foram ainda utilizados. Assim, embora estejam presentes nos vários processos, não são referidos na sua descrição.

Os vários elementos constituintes de um pacote (*containers*, tarefas, ciclos, etc.) possuem uma designação muitas vezes formada por um conjunto de palavras separadas por espaços. Para evitar qualquer confusão entre estas designações e o texto onde se inserem, estas são referidas neste documento entre aspas.

3.4.1 Processos de Extração (E)

Os processos de extração, tal como o nome indica, são aqueles que extraem os dados dos diversos sistemas operacionais do cliente e os colocam no *Operational Data Store* (ODS).

Nesta fase não existe qualquer tipo de transformação de dados, pois o seu objetivo é apenas a extração dos dados existentes, pois estes são necessários para a solução de BI, mantendo desta forma toda a lógica que está associada à solução de BI fora dos sistemas operacionais do cliente. Assim, como não existe qualquer tratamento de dados, tanto o modelo de indicadores como o modelo de qualidade seguem exatamente a mesma estrutura nos processos de extração.

Para a construção destes processos foram identificadas, em conjunto com o cliente, todas as tabelas e respetivas colunas que contêm a informação que é relevante para a solução de BI.

Para cada tabela e respetivas colunas identificadas, foi criada uma *view* no sistema fonte com o nome ODS_vwY, onde Y é o nome da tabela. Para cada uma destas *views* foi criado ainda um pacote de extração.

Todos os pacotes criados seguem uma sintaxe pré-definida o que permite uma rápida identificação de qual informação está a ser extraída, simplificando o entendimento de todos os membros da equipa, mantendo a coerência de todos no processo de desenvolvimento. A sintaxe pré-definida para estes pacotes é SSIS_E_X_Y, onde E identifica que é um pacote de Extração do *Integration Service*, X representa o nome da base de dados do cliente e Y representa o nome da *view* da qual os dados são extraídos.

Os 87 pacotes de extração criados podem ser divididos por três grupos de extratores, sendo estes definidos de acordo com o tipo de dados que extraem e pela lógica de extração inerente aos mesmos. Estes três grupos de extratores são os que geram as dimensões, os factos e as taxas de câmbio. Este último grupo é apenas constituído por um extrator que obtém informação proveniente de um *webservice*.

Processo de Extração de Dimensões

Os processos de extração de dados que originam as dimensões têm a estrutura observada na Figura 13.

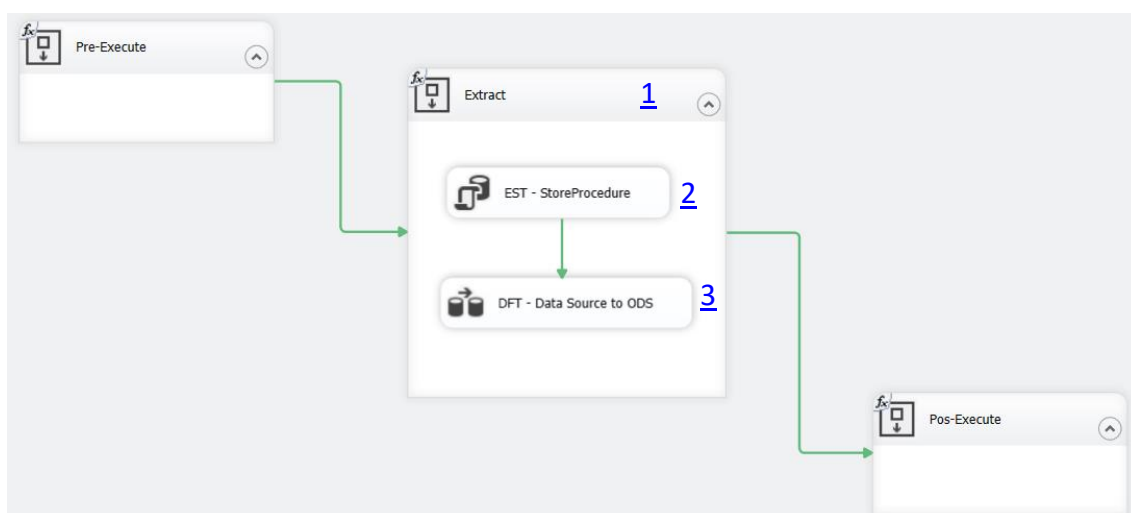


Figura 13 - Estrutura do processo de extração de dados para a criação de dimensões

Nesta figura estão identificados, de 1 a 3, os vários componentes envolvidos na extração. Estes componentes são:

- 1 “*Extract container*” – estrutura de execução que contém todas as operações envolvidas no processo de extração.
- 2 “*EST – StoreProcedure*” – tarefa que executa um procedimento criado na base de dados ODS que efectua o *truncate* da tabela de destino do processo de extração.
- 3 “*DFT – Data Source to ODS*” – tarefa que move os dados da respetiva *view* da fonte de dados e os transfere para a tabela destino no ODS.

Para tornar mais compreensível o processo de extração implementado, as tarefas “*EST – StoreProcedure*” e “*DFT – Data Source to ODS*” são seguidamente detalhadas.

A Figura 14 mostra a janela de configuração do “*EST – StoreProcedure*” (tarefa 2) e como são configurados os parâmetros necessários para que esta operação seja realizada com sucesso.

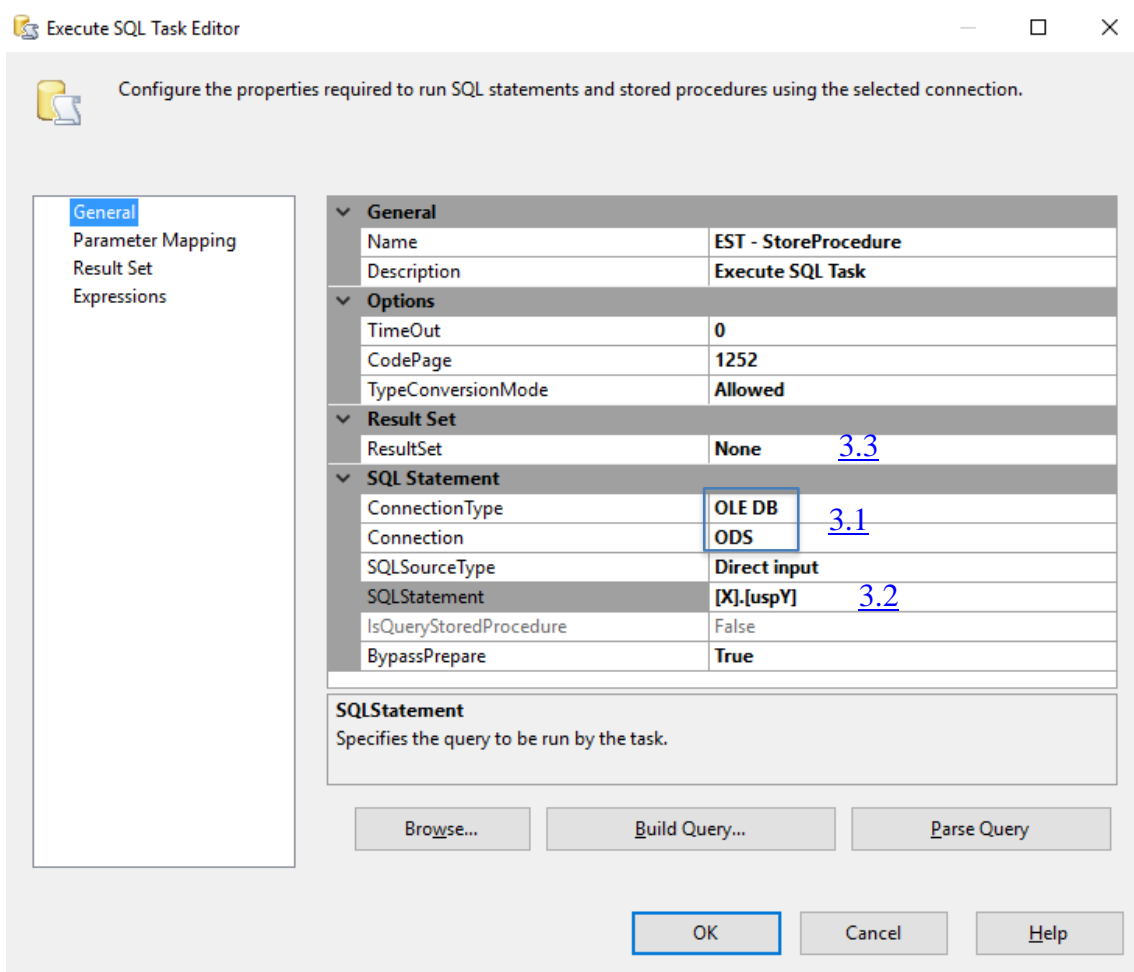


Figura 14 - Janela de configuração do *EST - StoreProcedure*

Neste caso a tarefa a efetuar é a execução de um procedimento existente na base de dados ODS que tem como objetivo realizar o *truncate* à respetiva tabela. Para tal, é necessário utilizar o *connection manager* criado inicialmente para o ODS (3.1) sendo assim possível fazer a chamada ao respetivo procedimento, tendo este o nome X.uspY (3.2), onde X representa o *schema* no ODS e que corresponde ao nome do sistema fonte dos dados, usp significa *user stored procedures* e é utilizado em todos os procedimentos criados ao longo da solução e Y representa o nome da *view* de onde os dados estão a ser extraídos. Como a tarefa a executar não retorna qualquer tipo de resposta, o seu *ResultSet* é colocado com o valor “None” (3.3).

A tarefa “DFT – Data Source to ODS” (tarefa 4) que tem como objetivo a extração dos dados da *view* do sistema fonte para o ODS, é composto pela sequência de atividades observadas na Figura 15. A primeira atividade é a OLE DB Source (4.1), que tem o nome ODS_vwY, onde Y representa o nome da respetiva *view*, e que é responsável por aceder à base dados do cliente e disponibilizar os dados existentes na *view*. A segunda atividade é a OLE DB Destination (4.2), que coloca os dados provenientes da atividade anterior na respetiva tabela X.Y do ODS, onde X representa o *schema* que tem o nome da fonte de dados e o Y representa o nome da tabela destino e que tem o mesmo nome da *view*.

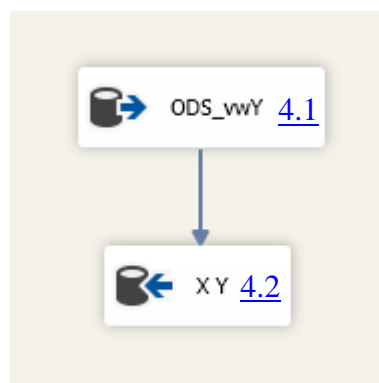


Figura 15 - Componentes envolvidos no passo de extração dos dados

Após a execução desta tarefa os dados já se encontram em ODS e o processo de extração dos dados que irão originar a respetiva dimensão encontra-se concluído. É possível observar que neste processo não existe nenhum mecanismo de filtragem de dados, isto porque estes são sempre extraídos na sua totalidade aquando da execução do processo diário.

Processo de Extração de Factos

Por outro lado, o mesmo já não é verificado para os processos de extração dos dados, que irão dar origem aos dados que irão popular as tabelas de factos. Este processo possui

maior complexidade e, por tratar uma quantidade considerável de dados, é necessário realizá-lo de forma incremental.

Para que a extração possa ser efetuada de forma incremental e dinâmica, contemplando períodos distintos para cada pacote, foi criada a tabela *DataSourceExtractionConfiguration* na base de dados de configuração, a qual tem a informação de cada pacote, onde se inclui a data de início e data de fim da extração. Todos estes processos têm um período de extração diário, com a exceção daqueles que dizem respeito aos indicadores de contabilidade que possuem um período de quatro meses.

A estrutura da tabela anteriormente referida é observada na Tabela 2.

Nome da coluna	Tipo de dados
PKDataSourceExtractConfiguration	Int
DataSource	Varchar(8000)
PackageName	Varchar(8000)
StartDate	Date
EndDate	Date

Tabela 2 - Estrutura da tabela *DataSourceExtractionConfiguration* da base de dados de configuração

Como pode ser observado nesta tabela, é guardada a informação de qual o sistema fonte a que o pacote vai extrair dados (*DataSource*), o nome do pacote (*PackageName*), a data de início (*StartDate*) e a data de fim (*EndDate*) do período de extração pretendido.

Com a tabela de configuração definida, foi então possível construir o pacote de extração de forma a que este tivesse em conta o respetivo período de extração. A Figura 16 representa a estrutura dos pacotes que são utilizados na extração de factos.

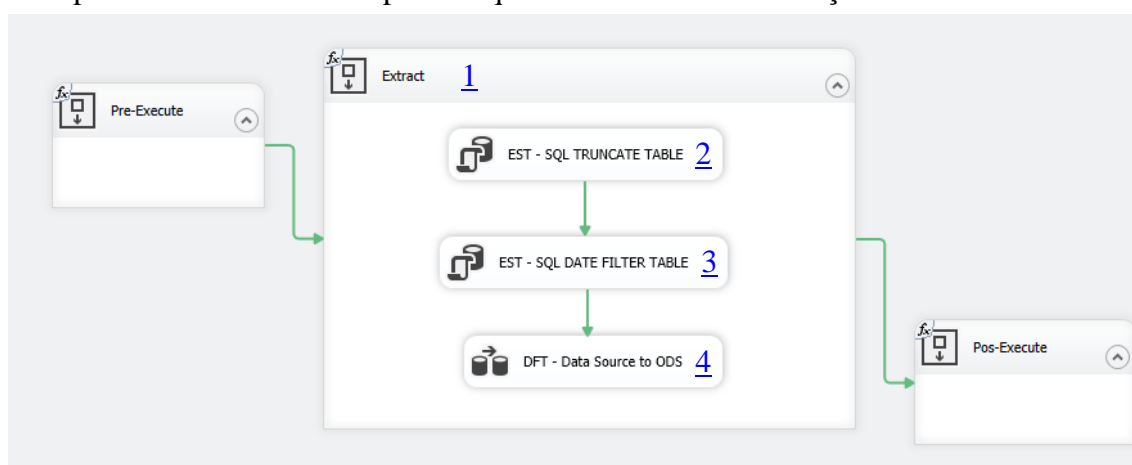
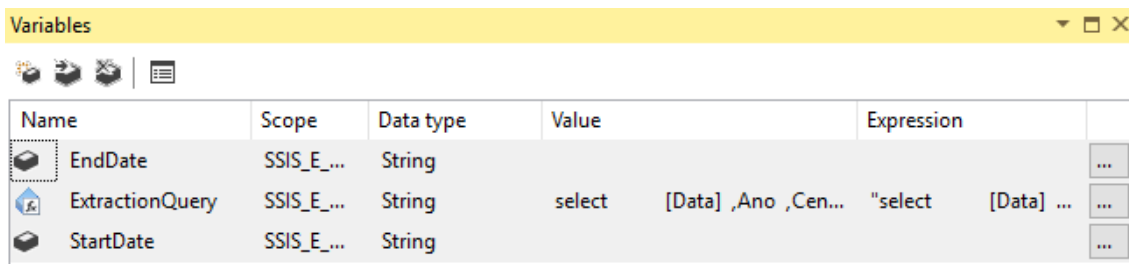


Figura 16 - Representação da estrutura do pacote de extração dos dados que originarão as tabelas de factos

Nesta figura estão identificados, de 1 a 4, os vários componentes envolvidos na extração. Estes componentes são:

1. “*Extract container*” – estrutura de execução que contém todas as operações envolvidas no processo de extração.
2. “*EST - SQL Truncate Table*” – tarefa que executa um procedimento criado na base de dados ODS que efetua o *truncate* da tabela de destino do processo de extração.
3. “*EST - SQL Date Filter Table*” – tarefa que executa um procedimento existente na base de dados ODS que recebe como parâmetro o nome do pacote e obtém como resultado a data de início e a data de fim do período de extração.
4. “*DFT – Data Source to ODS*” – tarefa que extrai os dados da respetiva *view* através da construção dinâmica de uma *query* de extração que inclui as datas recebidas da execução da tarefa anterior.

Tal como foi descrito, estes componentes utilizam as variáveis de data de início (*StartDate*), data de fim (*EndDate*) e a *query* de extração (*ExtractionQuery*). Assim, estas três variáveis têm de ser criadas antes da configuração dos componentes do *container* Extract. A Figura 17 mostra os diversos campos envolvidos na criação destas variáveis.



Name	Scope	Data type	Value	Expression
EndDate	SSIS_E...	String		
ExtractionQuery	SSIS_E...	String	select [Data] ,Ano ,Cen...	"select [Data] ...
StartDate	SSIS_E...	String		

Figura 17 - Vista das variáveis criadas no pacote de Integration Services

As variáveis *StartDate* e *EndDate* guardam a data de início e a data de fim do período de extração, respetivamente. Estes valores são devolvidos pelo procedimento executado pela tarefa “*EST - SQL Date Filter Table*” (tarefa 3). Estas variáveis são do tipo *String* devido ao *Regional Setting* da máquina virtual utilizada para o desenvolvimento, situada na *cloud*, ser diferente do ambiente de qualidade e de produção. Assim é garantido que não ocorrem erros de compatibilidade em qualquer uma das situações, nem é necessária a manutenção de versões distintas da aplicação para cada um dos ambientes.

A variável *ExtractionQuery*, também do tipo *String*, guarda a *query* que é executada na extração dos dados, mas com as duas variáveis referentes às datas de início e fim do período de extração incluídas na mesma, de forma a extrair apenas o período pretendido. Um exemplo de *query* de extração é o seguinte:

```
SELECT * FROM ODS_vwY WHERE Data between '"+ @[User::StartDate] +"' and '"+ @[User::EndDate] +'"
```

Neste exemplo o caracter *** é substituído pelas colunas pretendidas, *ODS_vwY* é o nome da respetiva *view*, *@[User::StartDate]* e *@[User::EndDate]* são substituídos, respetivamente, pelos valores que estas variáveis guardam .

Depois de criadas as variáveis, foi então configurada a tarefa “*EST - SQL Truncate Table*”, à semelhança da tarefa de extração “*EST – StoreProcedure*” descrita anteriormente. Esta tarefa chama um procedimento existente em ODS que faz o *truncate* à tabela destino da informação.

Segue-se a tarefa “*EST - SQL Date Filter Table*” (3) que consiste na execução de um procedimento que recebendo o nome do respetivo pacote devolve as suas datas de início e de fim do período de extração. A configuração deste componente é mostrada na Figura 18. Para a configuração do mesmo, é necessário estabelecer uma ligação à base de dados de configuração (4.1), executar o procedimento com o nome *uspReturnDatesToExtract* (4.2), que tem como parâmetro o nome do pacote (4.3), e finalmente receber o *ResultSet* com as respetivas datas (4.4) associando-as às respetivas variáveis (4.5).

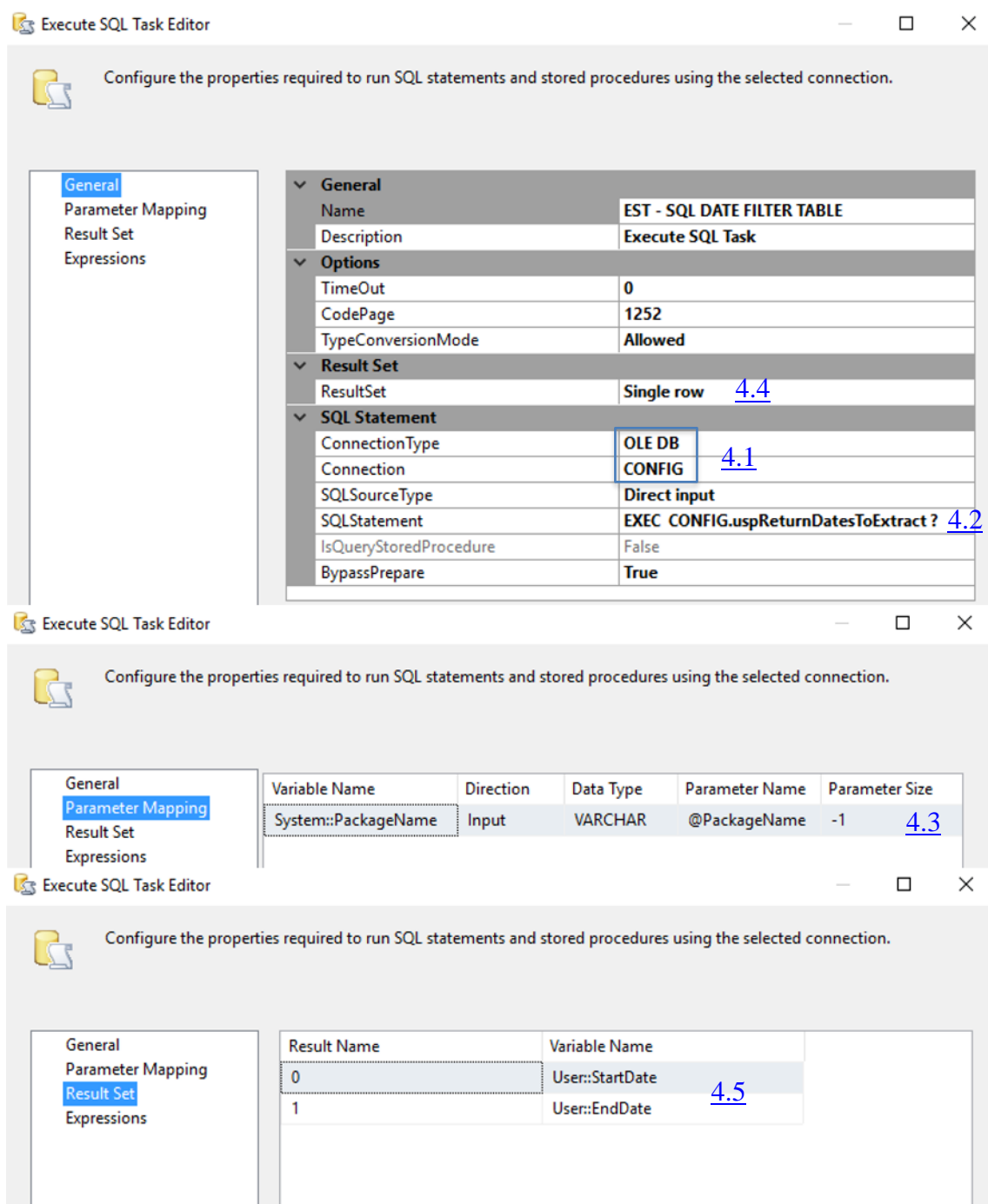


Figura 18 - Configuração do componente EST - SQL Date Filter Table

Com as variáveis de data de início e de fim do período de extração definidas, é agora possível executar o componente “DFT – Data Source to ODS” (5) o qual executa a *query* de extração que está na variável *Extraction Query*. Tal como referido anteriormente esta

query incorpora dinamicamente o valor das variáveis das datas de início e fim. Este passo encontra-se representado na Figura 19.

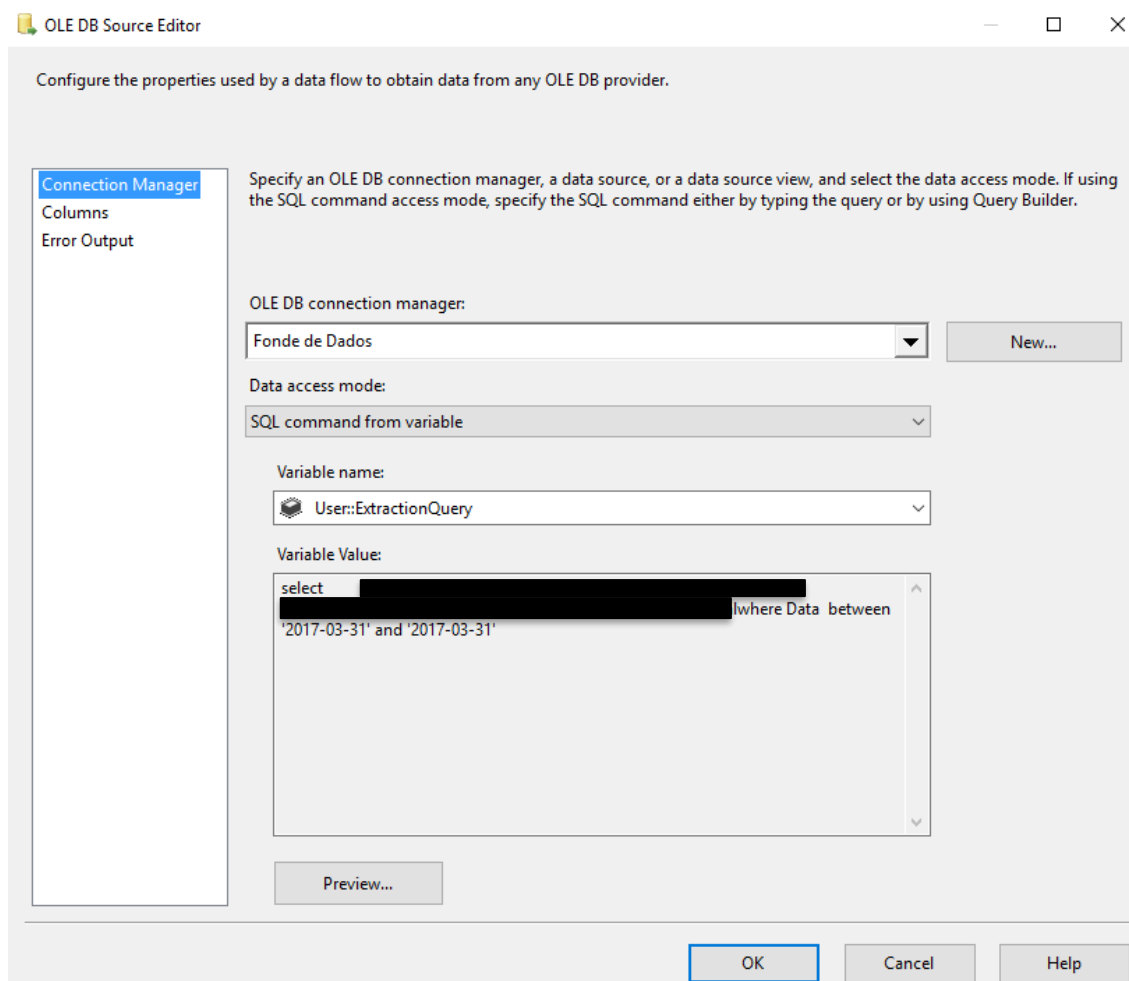


Figura 19 - Configuração do componente DFT – Data Source to ODS

À semelhança do descrito no processo de extração para as dimensões “*DFT – Data Source to ODS*”, após a execução desta *query* os dados são passados para a respetiva tabela em ODS, terminando assim o processo de extração de dados que irão originar factos.

Processo de extração de taxas de câmbio de WebService

Para concluir todos os processos de extração falta apenas o processo de que extrai, diariamente, as taxas de câmbio das moedas oficiais dos países onde o cliente tem operação. Para tal, foi construído um pacote com o nome *SSIS_E_WebService_ExchangeRate* que apresenta a estrutura que é apresentada na Figura 20.

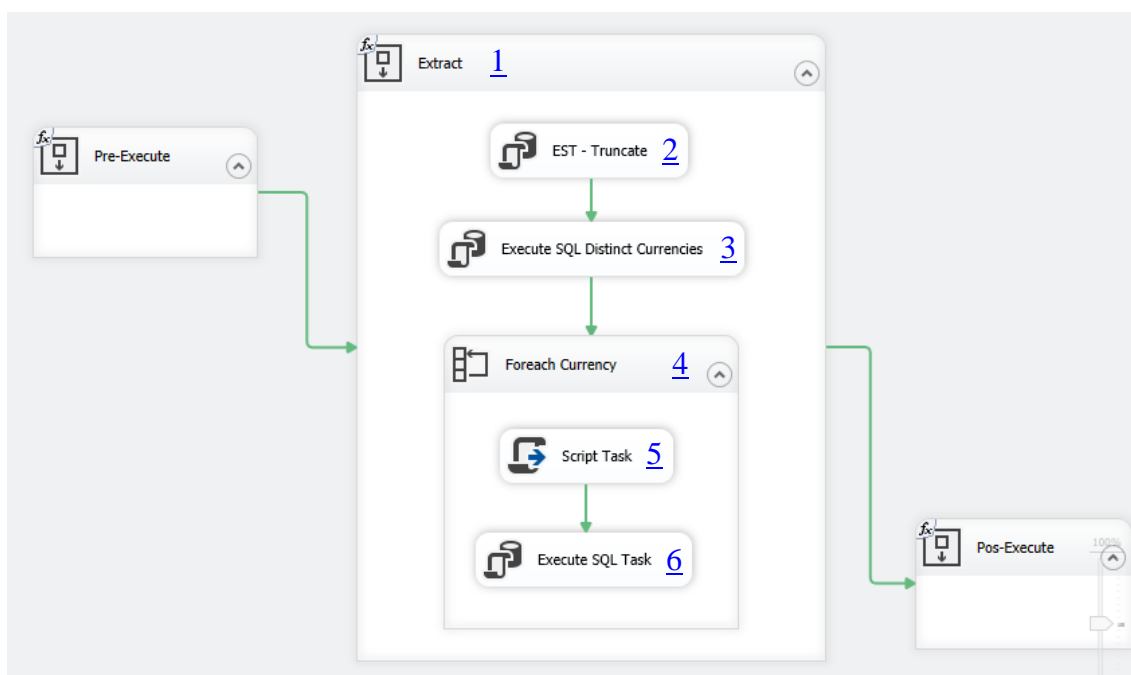
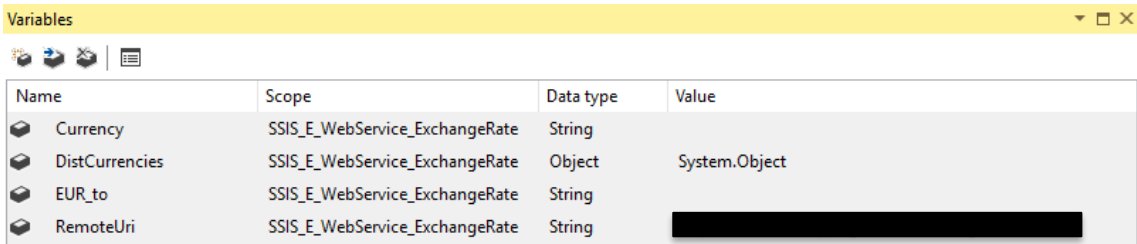


Figura 20 - Representação da estrutura do pacote SSIS_E_WebService_ExchangeRate

Neste pacote existem quatro variáveis e 6 tarefas, sendo estes:

1. “*Extract container*” – estrutura de execução que contém todas as operações envolvidas no processo de extração.
2. “*EST – Truncate*” – tarefa que executa um procedimento criado na base de dados ODS que efetua o *truncate* da tabela de destino do processo de extração.
3. “*Execute SQL Distinct Currencies*” – tarefa que executa um procedimento que devolve a lista de todas as moedas distintas que são utilizadas nos locais onde o cliente está presente.
4. “*Foreach Currency*” – ciclo que percorre a lista de moedas retornada na tarefa anterior e, para cada uma delas, executa as tarefas 5 e 6.
5. “*Script Task*” – tarefa que chama um *script* que para cada moeda faz um pedido ao *web service* e este retorna a taxa de câmbio dessa moeda relativamente ao euro.
6. “*Execute SQL Task*” – tarefa que executa um procedimento que insere na respetiva tabela de ODS a moeda em questão e o valor da taxa de câmbio daquele dia.

Para a criação deste processo foi necessário proceder à criação de 4 variáveis. São elas *Currency*, *DistCurrencies*, *EUR_to*, *RemoteURI*, que são mostradas na Figura 21.



Name	Scope	Data type	Value
Currency	SSIS_E_WebService_ExchangeRate	String	
DistCurrencies	SSIS_E_WebService_ExchangeRate	Object	System.Object
EUR_to	SSIS_E_WebService_ExchangeRate	String	
RemoteUri	SSIS_E_WebService_ExchangeRate	String	

Figura 21 - Representação das variáveis existentes no *SSIS_E_WebService_ExchangeRate*

A tarefa “*Execute SQL Distinct Currencies*” (3) devolve a lista das moedas que existem nas lojas e coloca-as na variável *DistCurrencies*. Posteriormente é feito um ciclo *for* (4) sobre esta lista, para cada moeda que se encontra lá presente (*Currency*), é feito um pedido ao *WebService*. Este pedido é feito através de um pequeno *script* desenvolvido em C# (6) que recebe a moeda (*Currency*) e o link para o *WebService* (*RemoteURI*), devolvendo o valor de câmbio do euro em relação a essa moeda para o respetivo dia e o colocando-o na variável *EUR_to*.

Por fim, a tarefa “*Execute SQL Task*” (6) consiste na execução de um procedimento que, para cada moeda, calcula a taxa de câmbio relativamente ao euro em falta, e insere estes valores na respetiva tabela em ODS.

3.4.2 Processos de Transformação e Carregamento (TL)

O objetivo destes processos é a realização das transformações que são necessárias efetuar sobre os dados extraídos, até que estes se encontrem com a estrutura final pretendida para alimentar a base de dados multidimensional.

Estes processos têm início com a construção de *views* na base de dados ODS, que são responsáveis pela normalização e otimização do tipo de dados e pelo tratamento de campos não preenchidos. Os dados resultantes destas *views* são então movidos para a base de dados de *Staging* onde ocorrem os restantes processos de transformação, sendo estes processos distintos para os dados que originam dimensões e para os que originam factos. Terminados os processos de transformação os dados são então carregados para a base de dados EDW, já com a estrutura final e prontos a alimentar a base de dados multidimensional.

À semelhança dos processos de extração, também nestes existem processos distintos para os dados que dão origem às dimensões e aos factos.

Tal como referido no início deste documento, são aqui abordados os processos para o modelo de qualidade e indicadores, dois dos modelos desenvolvidos para o cliente. É de salientar que os processos TL descritos para as dimensões e para os factos de qualidade têm a mesma estrutura e lógica de todos os outros modelos desenvolvidos na solução, sendo a única exceção o processo de TL dos factos para o modelo de indicadores, que tem um processo ligeiramente mais complexo.

Processos TL de Dimensões

Este processo é composto por dois *containers* principais, no primeiro é efetuada a transformação de dados e no segundo é efetuado o carregamento dos mesmos para a EDW.

À semelhança dos processos de extração, estes também são desenvolvidos em pacotes de *Integration Services* e possuem também uma sintaxe pré-definida, sendo esta SSIS_TL_X_DimY, onde X representa o *schema* e Y o nome da dimensão. A estrutura destes pacotes encontra-se representada na Figura 22.

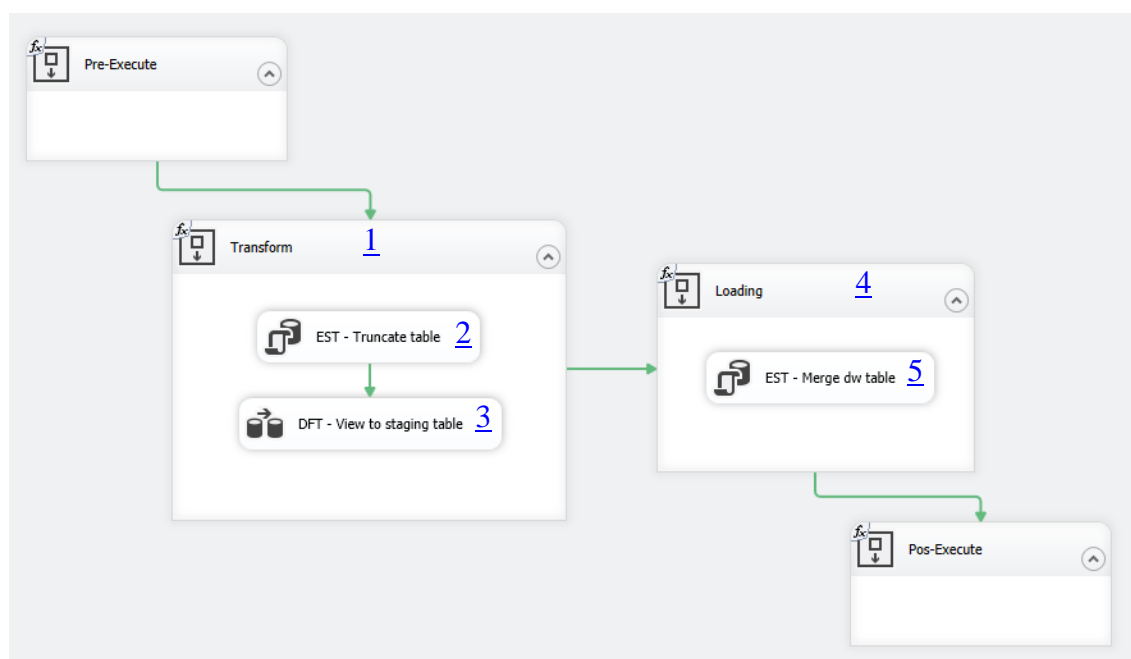


Figura 22 - Representação do processo de TL das dimensões

Tal como é visível na Figura 22 estes processos são compostos por cinco componentes, sendo estes:

1. “*Transform container*” – estrutura de execução que contém todas as operações envolvidas no processo de transformação dos dados.
2. “*EST – Truncate Table*” – tarefa que executa um procedimento criado na base de dados de *Staging* e que faz o *truncate* da tabela respetiva.

3. “*DFT – View to staging table*” – tarefa que move os dados da *view* existente em ODS para a tabela em *Staging*.
4. “*Loading container*” – estrutura que contém todas as operações envolvidas no carregamento dos dados.
5. “*EST – Merge dw table*” – tarefa que executa o procedimento criado na base de dados EDW que insere os novos dados e atualiza os dados já existentes, caso seja necessário, através do comando SQL *merge*.

Na etapa “*DFT – View to Staging Table*” (3) as transformações realizadas pelas *views* consistem essencialmente na normalização e otimização do tipo de dados, bem como o tratamento de campos não preenchidos.

As dimensões podem ter dois tipos distintos de comportamentos quando ocorrem atualizações dos dados já existentes. Assim e conforme o comportamento das mesmas estas dizem-se dimensões de tipo 1 ou tipo 2.

Quando uma dimensão apresenta um comportamento do tipo 1, isto significa que o histórico da informação que esta contém não é relevante. Assim, quando ocorrem alterações de dados basta fazer um *update* aos respetivos campos.

O mesmo já não é válido quando uma dimensão tem um comportamento do tipo 2. Neste caso o histórico da informação é relevante e a informação antiga tem de ser mantida porque não pode ser descartada. Assim, para todas as dimensões que têm este comportamento, são acrescentadas três colunas à tabela em EDW. Estas colunas são *validFrom*, *validTo* e *isActive*.

Sempre que uma linha nova é inserida, a coluna *validFrom* é preenchida com a data da inserção, a coluna *validTo* é preenchida com a data 9999-12-31 que representa o infinito e a coluna *isActive* é colocada com o valor *true*.

Quando ocorre uma alteração sobre uma linha já existente, esta passa a estar desatualizada, sendo a coluna *validTo* atualizada com a data atual, a coluna *isActive* é colocada com o valor *false* e é adicionada uma nova linha com a nova informação. Desta forma, com estas três colunas é possível saber se cada linha da dimensão está ativa e, no caso de já não estar, identificar o período de tempo em que a mesma esteve ativa. A Tabela 3 demonstra o comportamento descrito acima através de um pequeno exemplo onde o histórico da morada é relevante.

PK ^(*)	BK ^(**)	Nome	Morada	ValidFrom	ValidTo	isActive
1	0E984725-C51C-4BF4-9960-E1C80E27ABA0	João	Avenida de Ceuta	20/07/2016	15/06/2017	0
2	0E929475-C41C-4VGG4-9090-E1C80E27ABX5	Pedro	Avenida da Liberdade	20/07/2016	31/12/9999	1
3	0E984725-C51C-4BF4-9960-E1C80E27ABA0	João	Avenida de Roma	15/06/2017	31/12/9999	1

(*) Primary Key

(**) Business Key

Tabela 3 - Exemplo do comportamento de uma dimensão do tipo 2 com a ocorrência de um update

A tarefa “*EST – Merge DW Table*” (5) é responsável pela inserção e atualização dos dados existentes em EDW, para tal é utilizada a função *Merge* do SQL [16]. Este comando, que tem a estrutura apresentada na Figura 23, permite numa única operação realizar um *Insert* ou *Update* sobre a tabela de destino, tendo em consideração as condições verificadas na tabela de origem.

```

MERGE INTO [Target Table] Tgt
USING [Source Table] Src
ON (
    /*
     * Comparação dos campos chave da dimensão da tabela Source com a tabela Target
     */
)
WHEN MATCHED THEN
    /*
     * UPDATE
     */
WHEN NOT MATCHED THEN
    /*
     * INSERT
     */;

```

Figura 23 - Estrutura do comando Merge utilizado na inserção e atualização dos dados em EDW

No comando *merge* a tabela alvo é a tabela em EDW e a tabela fonte é a tabela de *Staging*. É verificado se a chave de negócio (BK) já existe na tabela de EDW e, caso exista, é feito o *update* da linha tendo em conta o tipo de dimensão. Caso a chave não exista, significa que é uma linha nova e é inserida.

Processos TL de Factos de Qualidade

Estes processos têm como objetivo a transformação dos dados de qualidade provenientes do ODS, até que estes possam ser utilizados no modelo multidimensional.

Assim, estes processos são responsáveis por realizar o mapeamento das chaves de negócio com as chaves utilizadas para as dimensões e proceder ao pré-cálculo de diversas métricas.

Tal como foi identificado no levantamento de requisitos, o modelo de qualidade incide sobre a avaliação da distribuição e a avaliação dos produtos. Como tal foram criados dois pacotes de IS que implementam os processos de TL respetivos. Estes pacotes foram denominados como *SSIS_TL_QUALITY_FactQualityAvDistribution* e *SSIS_TL_QUALITY_FactQualityAvProduct*.

No levantamento de requisitos foi ainda definido que cada avaliação pode ter mais do que um motivo associado. Isto implica a implementação de uma estrutura que permite a criação de uma relação muitos-para-muitos entre as avaliações e os motivos. Esta relação é conseguida com recurso a duas tabelas ponte (*Bridge Table* – BT), uma para cada um dos tipos de avaliação.

Foram então criados dois pacotes, com os seguintes nomes:

- *SSIS_TL_QUALITY_BTQualityAvDistributionReasons*
- *SSIS_TL_QUALITY_BTQualityAvProductReasons*.

A Figura 24 mostra como exemplo a estrutura do pacote *SSIS_TL_QUALITY_FactQualityAvDistribution* relativo ao processo de TL dos factos de avaliação da distribuição. A estrutura utilizada nos factos da avaliação dos produtos *SSIS_TL_QUALITY_BTQualityAvProductReasons* é em tudo idêntica.

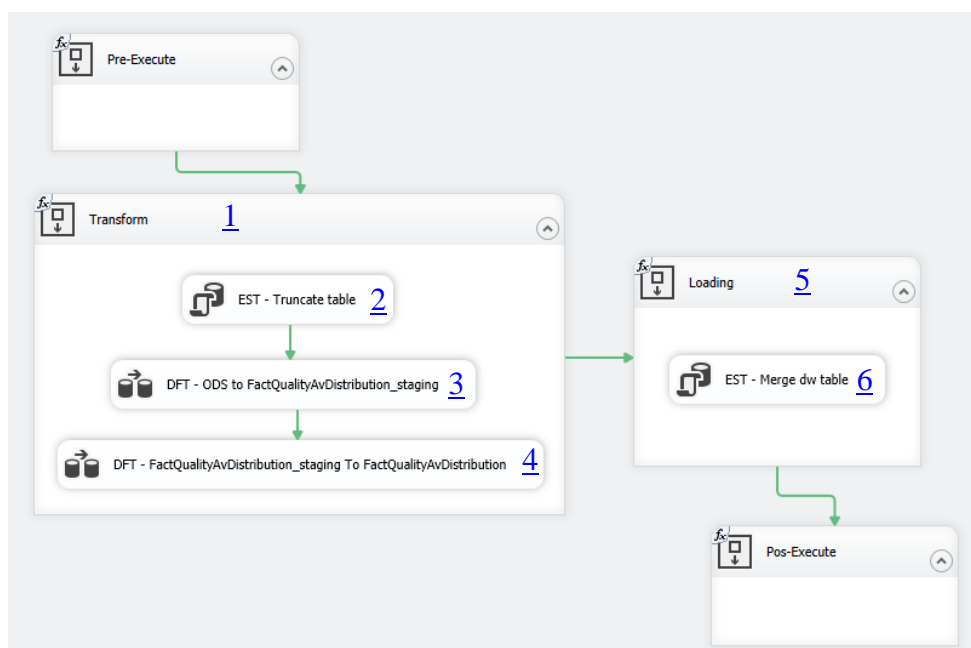


Figura 24 - Representação do processo de TL dos factos de qualidade – pacote *SSIS_TL_QUALITY_FactQualityAvDistribution*

Estes processos têm seis componentes, sendo estes:

1. “*Transform container*” – contém todas as operações envolvidas no processo de transformação dos dados.
2. “*EST – Truncate Table*” – tarefa que executa um procedimento criado na base de dados *staging* que faz o *truncate* às duas tabelas respetivas.
3. “*DFT – ODS to FactQualityAvDistribution_staging*” – tarefa que move os dados da *view* existente em ODS, que fornece os dados já, para a primeira tabela de *Staging*.
4. “*DFT – FactQualityAvDistribution_staging to FactQualityAvDistribution*” – tarefa que faz o mapeamento das chaves e pré-cálculo de métricas, e que move os dados de uma *view* construída sobre a primeira tabela de *staging* para a segunda.
5. “*Loading container*” – contém todas as operações envolvidas no carregamento dos dados.
6. “*EST – Merge dw table*” – tarefa que executa o *merge* sobre os dados dos factos. Se estes já existem em DW são atualizados caso tenham ocorrido alterações. Se estes não existirem são inseridos os novos dados.

Neste processo a tarefa “*DFT – ODS to FactQualityAvDistribution_staging*” (tarefa 3) é responsável por mover os dados tratados, provenientes da *view* de ODS, para a primeira tabela de *Staging*, que é denominada de *FactQualityAvDistribution_staging*. Sobre esta tabela é construída uma *view*, utilizada na tarefa “*DFT – FactQualityAvDistribution_staging to FactQualityAvDistribution*” (tarefa 4), que tem como funções realizar o mapeamento entre as chaves de negócio, que os dados trazem dos seus sistemas fonte, com as chaves utilizadas para cada dimensão e o pré-cálculo de algumas das métricas deste modelo.

Utilizando como exemplo a Tabela 3, pode-se verificar como é efetuado o mapeamento das chaves de negócio (BK) com a chave da dimensão (PK). Este mapeamento é feito essencialmente por dois motivos. O primeiro motivo, é que a utilização de uma chave de tipo inteiro é bastante mais rápida do que de outro tipo de dados. O segundo motivo, é que permite garantir que dados provenientes de fontes de dados distintas e que tenham a mesma chave de negócio não mapeiem a mesma chave de dimensão.

No mapeamento das chaves de negócio dos sistemas fonte com as chaves criadas para cada dimensão, existem três situações possíveis:

1. Existe relação entre a chave de negócio e a chave da dimensão;

2. Existe a chave de negócio, mas não existe relação com nenhuma chave da dimensão;
3. Não existe chave de negócio;

No primeiro caso, quando existem ambas as chaves o mapeamento é feito de forma direta, pois uma chave de negócio corresponde sempre a uma só chave de dimensão, no período em que esta está ativa.

No segundo e terceiro casos, o mapeamento não é possível porque, ou na dimensão ou na fonte de dados, não existe uma determinada chave. Nestes casos, o mapeamento é feito com duas chaves PK criadas especificamente para estas situações, sendo elas -1 e -2. Este mapeamento permite a fácil identificação de todas as linhas que falham alguma ligação com uma dimensão e automaticamente identifica o motivo pelo qual o mapeamento falhou.

Neste modelo são pré-calculadas três métricas que correspondem ao número de conformidades, ao número de não conformidades e ao número de devoluções.

Nos dados que chegam à tabela *FactQualityAvDistribution_staging*, a informação que permite o cálculo destas métricas encontra-se toda na mesma coluna, sendo que as avaliações conformes têm o valor C, as avaliações não conformes têm o valor N e as avaliações que levam a devoluções têm o valor D.

Para tornar o processo mais eficiente, a *view* cria três novas colunas as quais correspondem cada uma delas a um destes três tipos de avaliação. Depois, consoante a avaliação coloca o valor 1 na coluna correspondente e o valor 0 nas restantes. Este processo permite que na base de dados multidimensional o cálculo das métricas e seus agregados seja extremamente mais rápido, não ocorrendo assim nenhum tipo de demora nos processos de consulta.

Após este mapeamento e pré-cálculo, ocorre a tarefa “*DFT – FactQualityAvDistribution_staging to FactQualityAvDistribution*” (tarefa 4), que move esses dados da *view* para a tabela *FactQualityAvDistribution*, sendo que esta já tem a estrutura idêntica à respetiva tabela em EDW.

Para garantir que não existem dados duplicados em EDW, é executada a tarefa “*EST – Merge DW Table*” (tarefa 6) que executa o comando *merge* entre a tabela de *Staging* e a tabela de EDW, atualizando os dados já existentes ou inserindo apenas os novos dados caso estes ainda não existam.

Processos de T e TL de factos de indicadores

O processo de transformação e carregamento dos factos de indicadores é o processo mais complexo de todos os processos de ETL. Esta complexidade deve-se à forma como

têm de ser calculados e por terem de estar disponíveis nas várias moedas utilizadas nos países em que o cliente opera.

Os indicadores podem ser divididos em dois tipos: os simples e os complexos. Esta divisão é feita porque os indicadores simples são calculados sobre os dados extraídos para o ODS, enquanto que os indicadores complexos são calculados a partir dos indicadores simples. Esta dependência dos indicadores complexos, em relação aos indicadores simples, tem como implicação que o cálculo destes apenas seja realizado na base de dados multidimensional.

Estes processos que originam os factos dos indicadores estão divididos em dois passos. Um primeiro de transformação (T) e um segundo de transformação e carregamento (TL).

Por questões de eficiência, para melhorar o tempo de execução da solução como um todo, foi criado um processo de T para cada uma das áreas que têm indicadores, nomeadamente vendas, contabilidade, recursos humanos, qualidade, operacionais e administrativos. A opção de tornar o processo de transformação independente para cada uma das áreas previamente enunciadas, decorre do facto de cada um deles apresentar dependências distintas quando integrados no processo de execução diário. Assim, é possível paralelizar a execução de cada um tornando o tempo de execução de todo o processo diário mais curto. O processo diário será detalhado numa secção mais adiante deste trabalho.

O processo de TL deste modelo consolida o cálculo de todos os indicadores numa única tabela.

A Figura 25 apresenta a estrutura dos pacotes criados para o processo de T. Como exemplo é utilizado o processo de transformação que diz respeito aos indicadores de contabilidade.

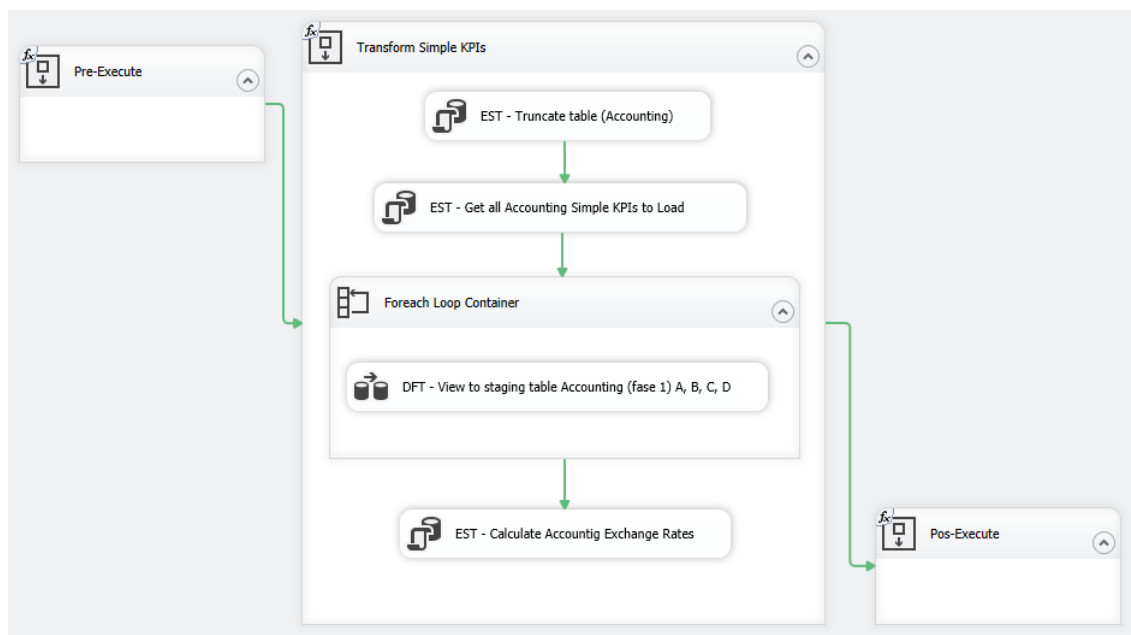


Figura 25 - Estrutura do processo T do modelo de indicadores

Este processo é composto por seis componentes, sendo estes:

1. “*Transform Simple KPIs container*” – contém todas as operações envolvidas no processo de transformação dos dados.
2. “*EST – Truncate table Accounting*” – tarefa que executa um procedimento criado na base de dados *staging* que faz o *truncate* à tabela respetiva.
3. “*EST – Get all Accounting Simple KPIs to load*” – tarefa que executa um procedimento que devolve todos os indicadores referentes à área de negócio que o pacote trata (no exemplo, é mostrado o caso da contabilidade).
4. “*Foreach Loop*” – ciclo que permite iterar sobre a lista de todos os indicadores de contabilidade existentes e para cada um deles executar a tarefa 5.
5. “*DFT – View to staging table Accounting fase 1 A, B, C, D*” – tarefa que calcula o valor de cada indicador para cada um dos cenários de análise possíveis utilizando cada uma das fórmulas obtidas na tarefa 3.
6. “*EST – Calculate Accounting Exchange Rates*” – tarefa que chama um procedimento da base de dados de *Staging* e que para todos os indicadores e

cenários volta a calcular o valor dos mesmos, mas convertendo os valores existentes para Euro.

Tal como foi anteriormente referido, após os processos de extração todos os dados que são utilizados no cálculo dos indicadores encontram-se já no ODS. Contudo, cada indicador simples tem uma fórmula de cálculo específica sobre os dados extraídos, sendo estas guardadas na tabela *KPIStatements* sob a forma de uma *query SQL*. A tarefa *EST – Get all Accounting Simple KPIs to load* (tarefa 3) é responsável por devolver todas as fórmulas dos indicadores, guardando-as numa lista.

O “*Foreach Loop container*” é responsável por percorrer todas as fórmulas existentes na lista permitindo assim a execução da tarefa “*DFT – View to staging table Accounting (fase 1) A, B, C, D*” (5) que sobre os dados do ODS aplica a fórmula do indicador e move o resultado para a respetiva tabela de *Staging*.

Na tarefa “*DFT – View to staging table Accounting (fase 1) A, B, C, D*” (tarefa 5) as letras A, B, C e D representam quatro dos oito cenários para os quais estes indicadores são calculados. No levantamento de requisitos foi identificado que os indicadores deveriam permitir análises em cenários monetários distintos, existindo oito cenários identificados pelas letras de A a H. Estes cenários são os seguintes:

- A – Actual Moeda Local – Análise efetuada tendo como base a Moeda local
- B – Actual Moeda Local – Análise de revisão do 4º Trimestre efetuada tendo como base a Moeda local
- C – Actual Moeda Local Ponderada – Análise ponderada efetuada tendo como base a Moeda local.
- D – Actual Moeda Local Ponderada – Análise ponderada de revisão do 4º Trimestre efetuada tendo como base a Moeda local.
- E – Actual Euro – Análise efetuada tendo como base o Euro
- F – Actual Euro – Análise de revisão do 4º Trimestre efetuada tendo como base o Euro
- G – Actual Euro Ponderada – Análise ponderada efetuada tendo como base o Euro.
- H – Actual Euro Ponderada – Análise ponderada de revisão do 4º Trimestre efetuada tendo como base o Euro.

Os cenários de A a D utilizam a moeda local como base de análise e, como tal são calculados sobre os dados provenientes dos sistemas fontes “*DFT – View to staging table Accounting (fase 1) A, B, C, D*” (tarefa 5). Os cenários de E a H utilizam o Euro como

moeda base, assim é necessário que após o cálculo dos cenários A a D os mesmos sejam convertidos do valor de moeda local para euro, à taxa de câmbio do dia em que os mesmos ocorreram, esta conversão é conseguida com a tarefa “*EST – Calculate Accounting Exchange Rates*” (tarefa 6).

Após a execução de cada um dos seis processos de T criados para os indicadores, é então executado o processo TL dos indicadores. Este processo é responsável por fazer o mapeamento das chaves de negócio para as chaves de dimensão correspondentes a cada uma das áreas de indicadores, juntar todas as áreas de indicadores numa única tabela com a estrutura final e, por fim, movê-las para EDW. A estrutura deste processo encontra-se apresentada na Figura 26.

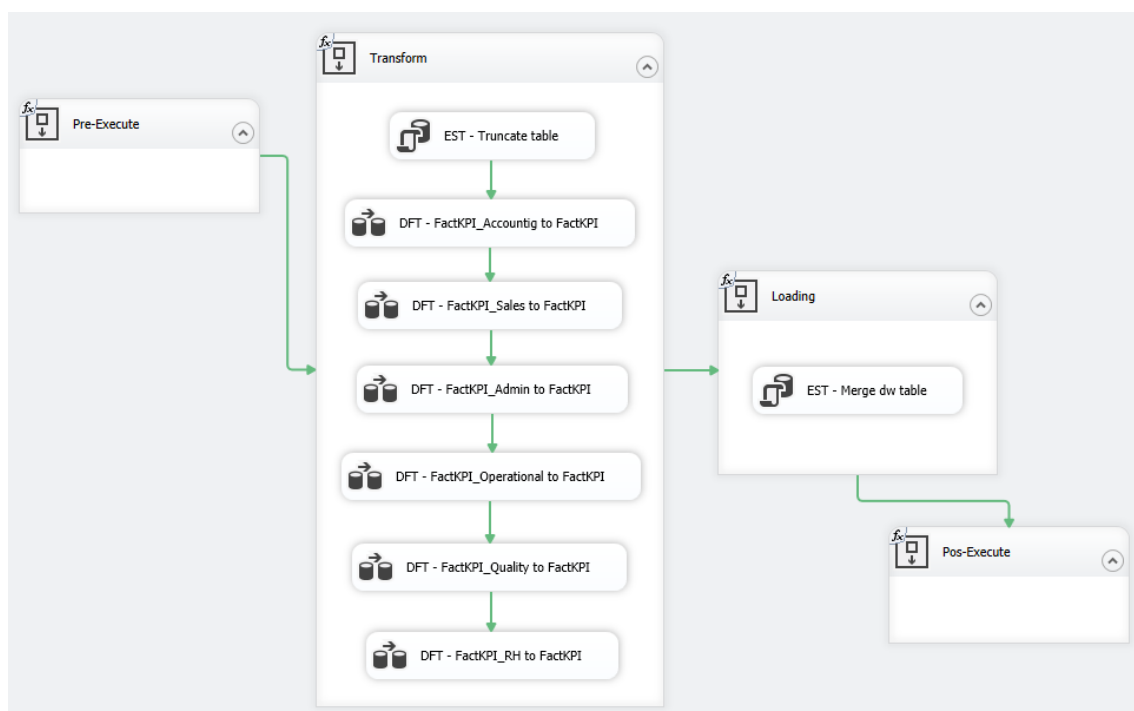


Figura 26 - Estrutura do processo TL do modelo de indicadores

Este processo é composto por dez componentes, sendo estes:

1. “*Transform container*” – contém todas as operações envolvidas no processo de transformação dos dados.
2. “*EST – Truncate Table*” – tarefa que executa um procedimento criado na base de dados *Staging*, que faz o *truncate* à tabela que contém todos os indicadores.
7. “*DFT – FactKPI_Accounting to FactKPI*” – tarefa que, através de uma *view*, faz o mapeamento das chaves com as dimensões que se encontram em EDW

e que move os dados que constrói sobre a tabela *FactKPI_Accounting* para a tabela de *staging FactKPI*.

8. “*DFT – FactKPI_Sales to FactKPI*” – tarefa que, através de uma *view*, faz o mapeamento das chaves com as dimensões que se encontram em EDW e que move os dados de uma *view* construída sobre a tabela *FactKPI_Sales* para a tabela de *staging FactKPI*.
3. “*DFT – FactKPI_Admin to FactKPI*” – tarefa que, através de uma *view*, faz o mapeamento das chaves com as dimensões que se encontram em EDW e que move os dados de uma *view* construída sobre a tabela *FactKPI_Admin* para a tabela de *staging FactKPI*.
4. “*DFT – FactKPI_Operational to FactKPI*” – tarefa que, através de uma *view*, faz o mapeamento das chaves com as dimensões que se encontram em EDW e que move os dados de uma *view* construída sobre a tabela *FactKPI_Operational* para a tabela de *staging FactKPI*.
5. “*DFT – FactKPI_Quality to FactKPI*” – tarefa que, através de uma *view*, faz o mapeamento das chaves com as dimensões que se encontram em EDW e que move os dados de uma *view* construída sobre a tabela *FactKPI_Quality* para a tabela *FactKPI* de *staging*.
6. “*DFT – FactKPI_RH to FactKPI*” – tarefa que, através de uma *view*, faz o mapeamento das chaves com as dimensões que se encontram em EDW e que move os dados de uma *view* construída sobre a tabela *FactKPI_RH* para a tabela *FactKPI* de *staging*.
9. “*Loading container*” – contém todas as operações envolvidas no carregamento dos dados.
7. “*EST – Merge DW Table*” – tarefa que executa o *merge* sobre os dados dos factos. Se estes já existem em DW são atualizados caso tenham ocorrido alterações. Se estes não existirem são inseridos os novos dados.

3.5 Base de Dados Multidimensional

Nesta solução foi utilizada uma base de dados multidimensional para suportar as diversas análises realizadas pelo cliente através dos seus *dashboards*.

Este tipo de base de dados foi escolhido por apresentar uma estrutura de dados que permite uma mais fácil compreensão e utilização dos mesmos e por apresentar um muito

melhor desempenho quando sujeita a *queries* complexas em comparação com uma base de dados relacional.

Para a construção desta base de dados e dos modelos multidimensionais da solução foi utilizado o *SQL Server Analysis Services* (SSAS).

3.5.1 Modelo de Indicadores

A construção do modelo implementado no cubo teve como ponto de partida o esquema criado no levantamento de requisitos (Figura 8), onde se encontram identificadas todas as dimensões utilizadas neste modelo. O passo seguinte foi ir adicionando a este modelo as estruturas necessárias para que, no final, todos os requisitos estivessem contemplados.

A primeira estrutura a ser implementada foi aquela que garante a segurança dinâmica dos dados, tendo em conta as permissões de cada utilizador, sabendo que estas permissões podem ser definidas de forma diferenciada para o acesso aos dados da loja e para o acesso aos dados dos indicadores.

Para se atingir este objetivo, foi definido que iria ser implementada uma segurança dinâmica dos dados sobre as dimensões[2][10]. Como tal, foram criadas duas tabelas ponte, uma para fazer a ligação entre a dimensão utilizador e a dimensão Loja e uma outra para fazer a ligação entre a dimensão utilizador e a dimensão Indicador. Estas tabelas ponte efetuam estes mapeamentos das permissões de uma forma totalmente independente. Assim, a estrutura do modelo, incluindo as alterações referidas, permite garantir que um determinado utilizador tem acesso apenas à(s) loja(s) e indicadores aos quais tem permissões, sendo que esta nova estrutura encontra-se representada na Figura 27.

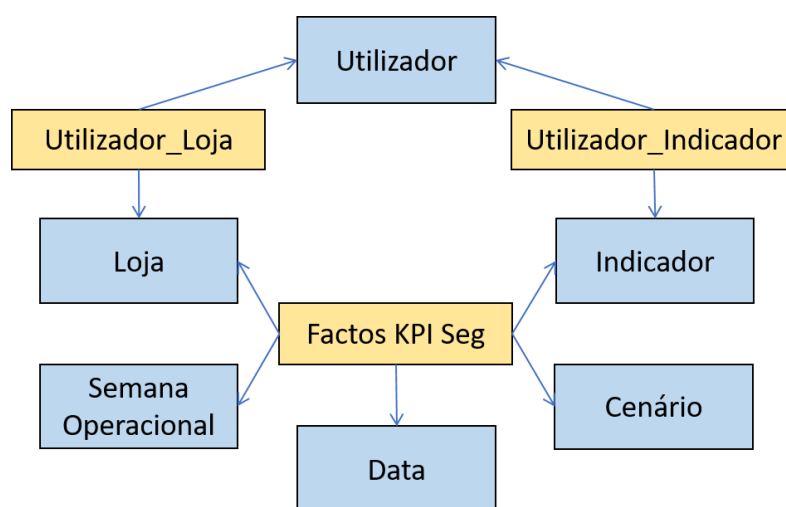


Figura 27 - Modelo de indicadores com o requisito de segurança implementado

A estrutura existente suporta o mecanismo de segurança pretendido, contudo este deve ser automático. Este automatismo tem como objetivo validar o utilizador que acede ao cubo com as suas permissões e confiná-lo apenas ao conjunto de dados ao qual pode ter acesso. A forma de realizar este processo é através da criação de um *Role*.

O *Analysis Services* fornece um modelo de acesso baseado em papéis (*roles*) que controla o acesso a todos os objetos e dados que nele se encontram. Para que o acesso seja autorizado, este tem que ser realizado no contexto de um *role* [17].

Como tal, foi criado um *Role* que tem como contexto todos os utilizadores, e que para a dimensão Loja e dimensão Indicador aplica código MDX (*Multidimensional Expressions*), que é a linguagem utilizada em *queries* a modelos multidimensionais. Este código MDX permite ao utilizador aceder aos dados quando a combinação chave primária (PK) e nome de utilizador existirem na respetiva tabela ponte. Estas configurações do *Role* são apresentadas na Figura 28, onde o código MDX apresentado é referente à dimensão Loja. O código referente à dimensão Indicador é em tudo semelhante.

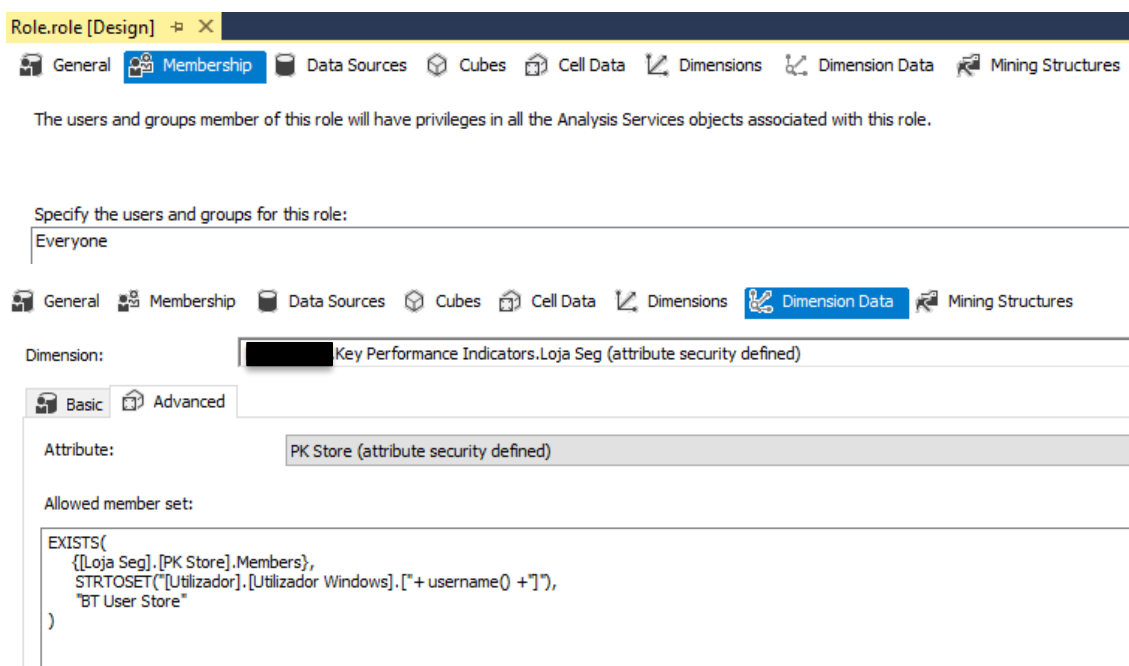


Figura 28 - Configuração da Role utilizada no modelo de indicadores

No levantamento de requisitos foi ainda identificado que o modelo tem que permitir o cálculo de diversos *rankings*, sendo estes calculados para cada indicador sobre um determinado universo de lojas, independentemente das permissões do utilizador que esteja a aceder. Contudo, com a estrutura do modelo definida, após a implementação do mecanismo de segurança (Figura 27) não é possível efetuar o cálculo destes *rankings* da forma pretendida.

O comportamento esperado seria o cálculo do *ranking* de cada loja para um determinado indicador e que após este cálculo fossem apenas apresentadas ao utilizador as lojas a que o mesmo tem acesso e o seu respetivo *ranking*. Porém este comportamento não se verifica, o *Analysis Services* primeiro restringe os dados que disponibiliza, consoante as *roles* definidas, e só depois é que efetua o cálculo dos *rankings*.

Como é imperativo que os *rankings* sejam calculados sem ter em conta as permissões do utilizador, ao mesmo tempo que este apenas tem acesso à informação a que as suas permissões o permitem, a estrutura utilizada no modelo teve de ser complementada.

A solução encontrada, que permite manter os requisitos de segurança e garantir o cálculo dos *rankings* da forma pretendida, passa por criar uma nova tabela de factos denominada de “Factos KPI”, em tudo igual à tabela “Factos KPI Seg”, com as mesmas ligações, e a esta última acrescentar uma nova ligação para a dimensão Loja e a dimensão Indicador. Estas alterações na estrutura do modelo são apresentadas na Figura 29.

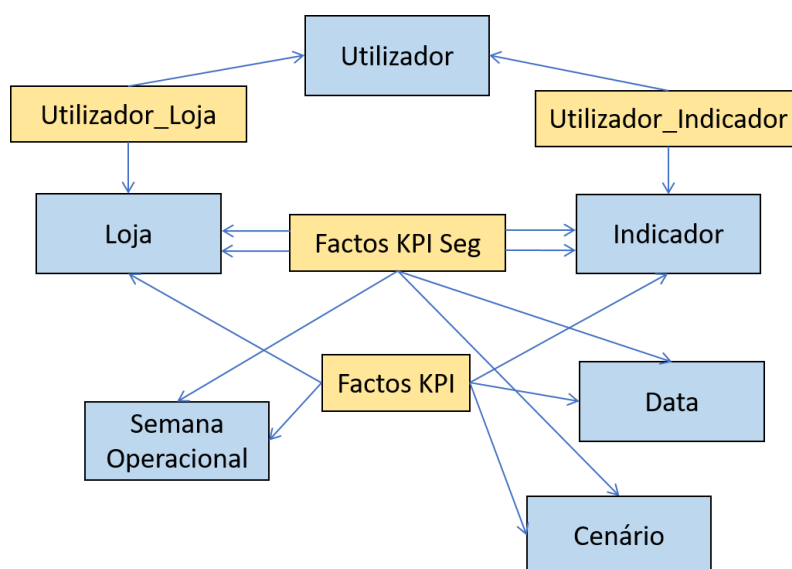


Figura 29 - Modelo implementado na construção do cubo dos indicadores

A mais do que uma ligação entre uma tabela de factos e uma dimensão, diz-se que existe um *role-playing*, o que significa que apesar de existir uma única tabela a nível físico, a mesma tem dois significados lógicos distintos [18].

Com esta estrutura definida é então possível responder aos requisitos de segurança e ao cálculo dos *rankings*.

A nova tabela de factos é exatamente igual à que já existia e é apenas utilizada para realizar o cálculo dos *rankings* por não ter qualquer mecanismo de segurança associado, permitindo que o cálculo seja sempre efetuado no universo completo de lojas.

As duas novas ligações da tabela de “Factos KPI Seg” para a dimensão Loja e dimensão Indicador criam, a nível lógico, duas novas dimensões distintas. Ficamos assim com a dimensão Loja e a dimensão Loja_segurança. O mesmo raciocínio é aplicado para a dimensão Indicador.

A dimensão Loja_segurança e Indicador_segurança fazem parte do mecanismo de segurança implementado e são elas que restringem os dados disponibilizados na tabela de “Factos KPI Seg”. Por outro lado, a dimensão Loja e a dimensão Indicador têm uma ligação a cada uma das tabelas de factos, sendo que todos os dados utilizados nas análises efetuadas provêm das tabelas “Factos KPI Seg” e “Factos KPI” que fornecem apenas os *rankings* calculados.

Com a estrutura do modelo definida, foi então possível fazer a modelação das dimensões, tabelas de factos e tabelas pontes utilizadas neste modelo. Neste processo de modelação, para cada uma das dimensões e tabelas foi definido o nome, uma breve descrição e o estado de visibilidade para cada coluna, sendo esta informação complementada com um exemplo do respetivo conteúdo.

A dimensão Data contém toda a informação temporal que permite efetuar as análises de negócio do cliente. A estrutura desta dimensão é apresentada na Tabela 4.

Nome da coluna	Descrição	Visível	Exemplo
Date Key	Chave primária	Não	20150101
Data Completa	Data completa	Não	2015-01-01
Dia	Dia do mês	Não	1
Semana	Semana do ano	Não	1
Mês	Número do mês	Não	1
Ano	Ano	Não	2015
Semestre	Semestre	Não	S1
Trimestre	Trimestre	Não	T1
Ano Semestre	Semestre do ano	Não	2015/S1
Ano Trimestre	Trimestre do ano	Não	2015/T1
Ano Mês	Mês do ano	Não	2015/01
Ano Semana	Semana do ano	Não	2015s01
Nome do Mês	Nome do mês	Sim	Janeiro
Dia da Semana	Dia da semana	Sim	Quinta-feira
É Dia de Semana	É fim de semana	Sim	True

Tabela 4 - Dimensão data

Sobre esta dimensão foram ainda criadas três hierarquias que permitem a análise dos dados com diferentes níveis de detalhe. Às mudanças no nível de detalhe chama-se *drill-down* e *roll-up*. O *drill-down* é quando é utilizada a hierarquia para obter mais detalhe sobre os dados, como por exemplo, passar de um período de análise anual para trimestral. O *roll-up* é exatamente o oposto, ou seja, é quando se pretende obter uma vista mais generalizada dos dados passando, por exemplo, de uma análise trimestral para uma análise anual. As três hierarquias criadas encontram-se representadas na Figura 30.

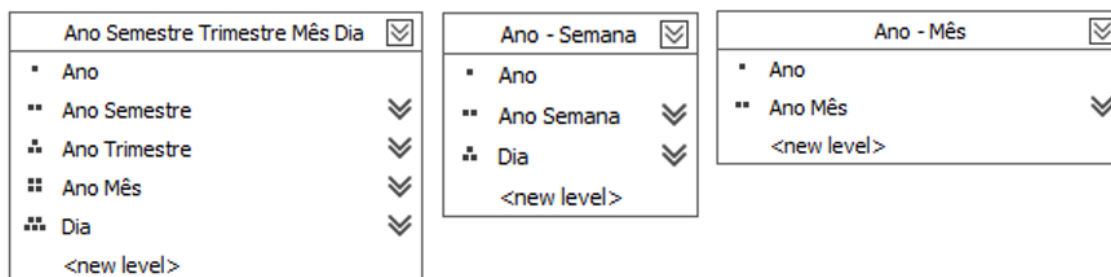


Figura 30 - Representação das hierarquias da dimensão data

A dimensão Loja contém toda a informação relativa ao detalhe de uma loja. Embora existam duas representações lógicas desta dimensão neste modelo, a estrutura aqui apresentada é igual em ambas. Relativamente à visibilidade destas, a dimensão Loja_segurança não está visível, pois apenas a dimensão Loja está acessível ao utilizador. A estrutura desta dimensão está representada na Tabela 5.

Nome da coluna	Descrição	Visível	Exemplo
PKStore	Chave primária	Não	1
Código	Código identificativo da loja	Sim	001
Código Postal	Código postal	Sim	2645-543
Conceito	Conceito em que a loja se enquadra	Sim	Shopping
Data de abertura	Data de abertura da loja	Sim	2015-02-01
Data de criação	Data de criação da loja	Sim	2015-01-01
Data de encerramento	Data de fecho da loja	Sim	2017-01-01
Designação postal	Nome da localidade do código postal	Sim	Alcabideche
Distrito	Distrito em que a loja se insere	Sim	Lisboa
Email	Email da loja	Sim	loja1@empresa.com
Empresa	Empresa da Loja	Sim	Empresa 1
Esquema da loja	Disposição física da loja	Sim	1

Franchise	Identifica se a loja é um <i>franchise</i> ou não	Sim	1
Localidade	Localidade	Sim	Alcabideche
Loja Ativa	Identifica se a loja está ativa na atualidade	Sim	1
Moeda	Moeda utilizada	Sim	EUR
Morada	Morada da loja	Sim	CascaShopping
Nome	Nome da loja	Sim	Loja 1
Número de POS	Número de pontos de venda da loja	Sim	3
Número Fiscal	Número fiscal	Sim	123456789
País	País	Sim	Portugal
Região	Região	Sim	Cascais
Supervisor	Nome do supervisor da loja	Sim	João Silva
Telefone	Contacto telefónico	Sim	211231234
Grupo 1	Campos utilizados para o cálculo dos <i>rankings</i>	Sim	*
Grupo 2		Sim	*
Grupo 3		Sim	*
Grupo 4		Sim	*
Grupo 5		Sim	*
Grupo 6		Sim	*
Grupo 7		Sim	*
Grupo 8		Sim	*
Grupo 9		Sim	*
Grupo 10		Sim	*
BKCode	Chave de negócio que identifica a loja	Não	4J7647565-JUCR-4BF4-9960-E1LK947JABL3
BKCompany	Chave de negócio que identifica a empresa a que a loja pertence	Não	9R984725-DF58C-4GF7-9361-H1C80T32AFS7
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

* Estes campos podem conter qualquer valor desde que suportado pelo tipo de dados *varchar(50)*

Tabela 5 - Dimensão Loja

Os campos Grupo 1 a Grupo 10 presentes na dimensão são utilizados para o cálculo dinâmico dos *rankings*, sendo que é com o conteúdo destas colunas que é definido o universo de análise de lojas sobre o qual o *ranking* e as métricas devem ser calculados utilizando MDX no cubo. Este mecanismo será detalhado mais adiante.

Esta dimensão contém a informação referente aos indicadores e, à semelhança da dimensão Loja, também esta estrutura é igual na dimensão Indicador_segurança e na dimensão Indicador. A dimensão Indicador_segurança não está visível para o utilizador, tendo este apenas acesso à dimensão Indicador. A estrutura desta dimensão está representada na Tabela 6.

Nome da coluna	Descrição	Visível	Exemplo
PKKPI	Chave primária	Não	1
Indicador	Nome do indicador	Sim	Custo de venda
Grupo	Área de negócio à qual o indicador pertence	Sim	Contabilidade
Tipo	Campo que identifica o tipo de indicador (quantidade, moeda ou percentagem)	Sim	Moeda
RankType	Campo que identifica a forma como o <i>ranking</i> é calculado. <i>RankType</i> = 1 quanto maior o valor melhor o <i>ranking</i> <i>RankType</i> = -1 quanto menor o valor melhor o <i>ranking</i>	Não	-1
BKKPI	Chave de negócio que identifica o indicador	Não	CO002
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 6 - Dimensão Indicador

Para esta dimensão foram ainda definidas uma hierarquia entre o grupo e o indicador, que se encontra representada na Figura 31.

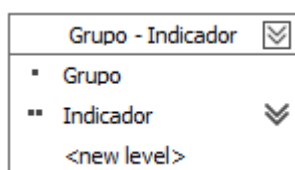


Figura 31 - Hierarquia da dimensão indicador

A dimensão Utilizador contém a informação relativa aos utilizadores que têm acesso à solução e a sua estrutura é representada na Tabela 7. Esta dimensão não se encontra visível para o utilizador que acede à solução.

Nome da coluna	Descrição	Visível	Exemplo
PKUser	Chave primária	Não	1
BKUser	Chave de negócio que identifica o utilizador	Não	2T529525-K65TC-29Y7-AF62-L9I86F8SFFS6
Nome	Nome do utilizador	Não	João
WindowsUsername	Conta que o utilizador usa para fazer <i>login</i> no sistema	Não	Joao@empresa.com
Cargo	Cargo que o utilizador desempenha	Não	Administração
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 7 - Dimensão Utilizador

A dimensão Semana Operacional contém a informação referente ao período de operação de cada uma das empresas do cliente, podendo, para uma mesma data, cada uma das empresas estar em períodos de operação distintos.

Nome da coluna	Descrição	Visível	Exemplo
PKOperationalWeek	Chave primária	Não	1
BKOperationalWeek	Chave de negócio que identifica a semana operacional	Não	2T529525-K65TC-29Y7-AF62-L9I86F8SFFS6
BKCompany	Chave de negócio que identifica a empresa	Não	9R984725-DF58C-4GF7-9361-H1C80T32AFS7
BKYear	Chave de negócio que identifica o ano	Não	2WE59525-L67NC-29Y7-AF62-L9JDL85S20J6
Semana Operacional	Valor numérico que identifica a semana do ano referida (1 a 52)	Sim	1
Empresa	Nome da empresa		Empresa 1
Início da semana			07-01-2015
Fim da semana		Não	14-01-2015
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 8 - Dimensão semana operacional

A tabela ponte entre a dimensão Utilizador e a dimensão Loja permite o mapeamento entre os utilizadores e as lojas para as quais têm permissões de acesso através da utilização das chaves primárias de cada uma. Esta tabela não se encontra visível para o utilizador e a sua estrutura está representada na Tabela 9.

Nome da coluna	Descrição	Visível	Exemplo
FKUser	Chave estrangeira da chave primária do utilizador	Não	1
FKStore	Chave estrangeira da chave primária da loja	Não	5

Tabela 9 - Tabela ponte entre a dimensão Utilizador e a dimensão Loja

A tabela ponte entre a dimensão Utilizador e a dimensão Indicador permite o mapeamento entre os utilizadores e os indicadores para os quais têm permissões de acesso através da utilização das chaves primárias de cada uma. Esta tabela não se encontra visível para o utilizador e a sua estrutura está representada na Tabela 10.

Nome da coluna	Descrição	Visível	Exemplo
FKUser	Chave estrangeira da chave primária do utilizador	Não	1
FKKPI	Chave estrangeira da chave primária do indicador	Não	5

Tabela 10 - Tabela ponte entre a dimensão Utilizador e a dimensão Indicador

As tabelas de “Factos KPI Seg” e a “Factos KPI” contêm a mesma informação, sendo que a única diferença estrutural é a tabela “Factos KPI Seg” possuir duas colunas extra, que são utilizadas na ligação à dimensão Loja e dimensão Indicador utilizadas no mecanismo de segurança. A tabela de “Factos KPI Seg” é a que se encontra visível e à qual o utilizador tem acesso, a tabela “Factos KPI” é invisível para o utilizador.

A estrutura destas tabelas encontra-se representada na Tabela 11.

Nome da coluna	Descrição	Visível	Exemplo
PKFact	Chave primária do facto	Não	1
FKStore	Chave estrangeira da chave primária da loja	Não	5
FKKPI	Chave estrangeira da chave primária do indicador	Não	5
FKDate	Chave estrangeira da chave primária da data	Não	20150102
FKScenario	Chave estrangeira da chave primária do cenário	Não	3
FKOperationalWeek	Chave estrangeira da chave primária da semana operacional	Não	2
Valor Atual	Valor do indicador realizado pela loja naquele dia e no respetivo cenário	Sim	1000

Valor Orçamentado	Valor do indicador esperado realizar pela loja naquele dia e no respetivo cenário	Sim	750
-------------------	---	-----	-----

Tabela 11 - Tabela de factos do modelo de indicadores

Com a estrutura do modelo e de todas as dimensões, tabelas ponte e tabelas de factos foi então construído o cubo utilizado para analisar o modelo de indicadores.

Contudo, para que o cubo fique completo, é necessário ainda acrescentar as métricas calculadas, que foram identificadas no levantamento de requisitos, bem como o cálculo dos indicadores complexos detetados no processo de ETL. Todas estas métricas foram calculadas em MDX.

O cálculo dos indicadores complexos utiliza a função *scope* do MDX, que no fundo permite executar um determinado pedaço de código quando aquele objeto específico é solicitado. Na Figura 32 temos dois exemplos de cálculo dos indicadores com o código CO003 e CO004.

```

SCOPE ([Indicador].[Indicador].&[CO003]);
    THIS = IIF([Indicador].[Indicador].&[CO001] = 0 or [Indicador].[Indicador].&[CO001] is null,
        null, [Indicador].[Indicador].&[CO002]/[Indicador].[Indicador].&[CO001]);
    FORMAT_STRING(THIS) = "#.##%";
END SCOPE;

SCOPE ([Indicador].[Indicador].&[CO004]);
    THIS = [Indicador].[Indicador].&[CO001]-[Indicador].[Indicador].&[CO002];
END SCOPE;

```

Figura 32 - Código MDX utilizado no cálculo dos indicadores complexos

No primeiro caso, o indicador CO003 é calculado a partir da divisão do indicador CO002 pelo indicador CO001 e é esta fórmula que é executada sempre que o indicador CO003 é calculado. No segundo caso o indicador CO004 é calculado efetuando a subtração do valor do indicador CO002 ao valor do indicador CO001.

Outras métricas que são também necessárias calcular são o período homólogo, a percentagem do valor realizado face ao valor orçamentado, a percentagem do valor realizado face ao valor homólogo, a diferença entre o valor realizado e o valor homólogo e ainda a diferença entre o valor realizado e o valor orçamentado. A fórmula utilizada para o cálculo destas encontra-se na Figura 33.

```

CREATE MEMBER CURRENTCUBE.[Measures].[Homólogo]
AS (ParallelPeriod([Data].[Ano Semestre Trimestre Mês Dia].[Ano],
1,
[Data].[Ano Semestre Trimestre Mês Dia].CurrentMember)
,[Measures].[Realizado]),
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'KPI';

CREATE MEMBER CURRENTCUBE.[Measures].[R/O %]
AS IIF(IsEmpty([Measures].[Orçamento]), null,
([Measures].[Realizado] / [Measures].[Orçamento])-1),
FORMAT_STRING = "Percent",
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'KPI';

CREATE MEMBER CURRENTCUBE.[Measures].[R/H %]
AS IIF(IsEmpty([Measures].[Homólogo]), null,
([Measures].[Realizado] / [Measures].[Homólogo])-1),
FORMAT_STRING = "Percent",
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'KPI';

CREATE MEMBER CURRENTCUBE.[Measures].[R-O]
AS [Measures].[Realizado] - [Measures].[Orçamento],
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'KPI';

CREATE MEMBER CURRENTCUBE.[Measures].[R-H]
AS [Measures].[Realizado] - [Measures].[Homólogo],
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'KPI';

```

Figura 33 - Código MDX para o cálculo do período homólogo, percentagem entre o valor realizado e o orçamentado, percentagem entre o valor realizado e o homólogo, diferença entre o valor realizado e o orçamentado e a diferença entre o valor realizado e o homólogo

Com o cálculo dos indicadores complexos, do conjunto de métricas descritos acima só falta o cálculo do número de lojas, a média e o *ranking* para cada um os dez conjuntos de métricas dinâmicas.

A Figura 34 apresenta o código MDX utilizado no cálculo do *ranking*, *ranking* com total, número de loja e média.

```

/* ***** Grupo 1 / Conceito ***** */
CREATE MEMBER CURRENTCUBE.[Measures].[RankConceito]
AS IIF(IsEmpty([Measures].[Realizado]), null,
RANK((([Loja].[Nome].CurrentMember, strtomember(' [Loja].[Grupo1]&'+[Loja].[Nome].CurrentMember.properties('Grupo1')+')),
((FILTER([Loja].[Nome].[Nome], ([Indicador].[Indicador].&[VD001], [Measures].[Value]))),
strtomember(' [Loja].[Grupo1]&'+[Loja].[Nome].CurrentMember.properties('Grupo1')+')),
([Measures].[Value]*[Indicador].[Indicador].properties( "Rank Type" ))))),
VISIBLE = 1 , DISPLAY_FOLDER = 'Grupo de análise\Conceito' , ASSOCIATED_MEASURE_GROUP = 'KPI' ;

CREATE MEMBER CURRENTCUBE.[Measures].[RankConceitoComTotal]
AS IIF(IsEmpty([Measures].[RankConceito]), null,
(CSTR([Measures].[RankConceito]) + ' / ' + CSTR([Measures].[NumLojasConceito] )))
, VISIBLE = 1 , DISPLAY_FOLDER = 'Grupo de análise\Conceito' , ASSOCIATED_MEASURE_GROUP = 'KPI';

CREATE MEMBER CURRENTCUBE.[Measures].[NumLojasConceito]
AS IIF(IsEmpty([Measures].[Realizado]), null,
count(nonempty(crossjoin([Loja].[Grupo1].currentmember, [Loja].[Nome].[Nome], [Indicador].[Indicador].&[VD001], [Measures].[Value]))))
, VISIBLE = 1 , DISPLAY_FOLDER = 'Grupo de análise\Conceito' , ASSOCIATED_MEASURE_GROUP = 'KPI';

CREATE MEMBER CURRENTCUBE.[Measures].[MediaConceito]
AS IIF(IsEmpty([Measures].[Realizado]), null,
sum((((FILTER([Loja].[Nome].[Nome], ([Indicador].[Indicador].&[VD001], [Measures].[Value])
, strtomember(' [Loja].[Grupo1]&'+[Loja].[Nome].CurrentMember.properties('Grupo1')+')), [Measures].[Value]))
/[Measures].[NumLojasConceito]
, VISIBLE = 1 , DISPLAY_FOLDER = 'Grupo de análise\Conceito' , ASSOCIATED_MEASURE_GROUP = 'KPI';

```

Figura 34 - Código MDX utilizado no cálculo do número de lojas, média e ranking

Neste caso, o exemplo utilizado é o cálculo do *ranking* relativo ao conceito da loja. Os universos de análise do conceito são definidos na coluna “Grupo 1” da dimensão loja, tendo cada universo tem o seu respetivo valor. No caso de a coluna possuir dois ou mais valores distintos são calculados dois ou mais *rankings*, um para cada um desses valores que representam o *ranking* da loja no conceito em que se insere.

Esta lógica é aplicada em todos os cálculos dos *rankings* e médias, cada um baseado na sua respetiva coluna de Grupo da dimensão Loja.

Com a estrutura do modelo definida permitindo a resposta a todos os requisitos e métricas identificadas pelo cliente o desenvolvimento do modelo de indicadores foi terminado ficando pronto a ser consultado.

3.5.1 Modelo de Qualidade

A construção do modelo de qualidade implementado no cubo teve início com o desenho do modelo analítico que o ia suportar. O ponto de partida para a construção deste modelo, foi a utilização dos esquemas criados no levantamento de requisitos para a avaliação da distribuição (Figura 9) e para a avaliação dos produtos (Figura 10 **Error! Reference source not found.**).

Embora tenham sido criados estes dois esquemas durante o processo de levantamento de requisitos para o modelo de qualidade, estes foram construídos apenas para simplificar a definição destas duas vertentes de avaliação junto do cliente.

A construção do modelo de qualidade origina apenas um único cubo, que permite a análise destes dois tipos de avaliações. Isto só é possível por existirem dois grupos de métricas distintos, um para cada tipo de avaliação. Assim, os esquemas criados no levantamento de requisitos foram fundidos, começando deste modo a estruturar o modelo implementado. O esquema que surgiu desta união está representado na Figura 35.

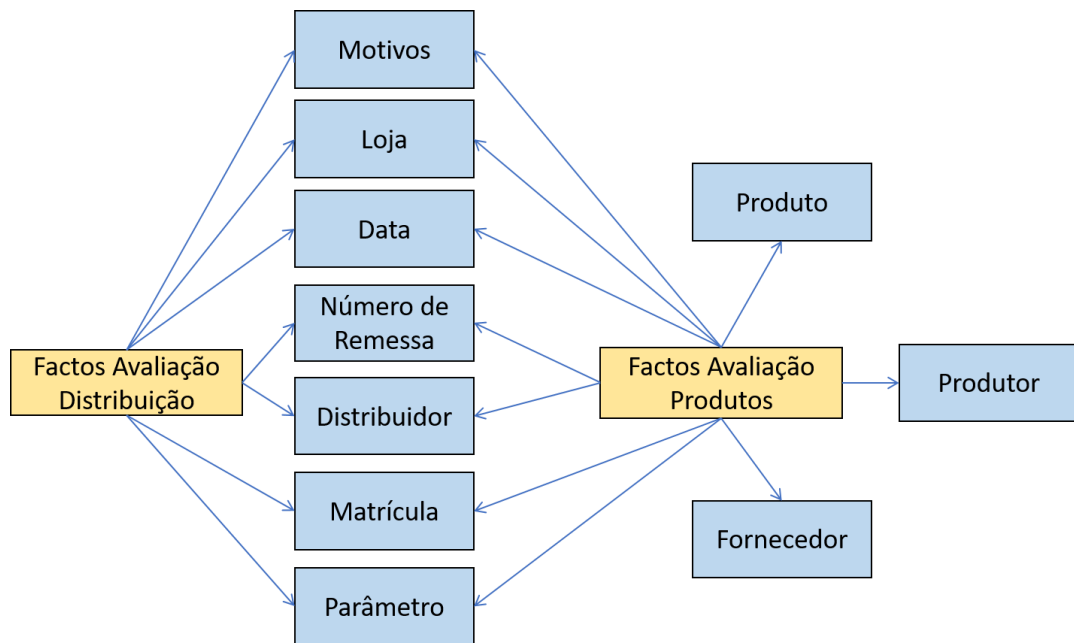


Figura 35 - Esquema do modelo de qualidade incluindo os dois tipos de avaliação

Confrontando o esquema da Figura 35 com os requisitos identificados, este ainda não permite dar resposta ao caso de uma avaliação poder ter múltiplos motivos associados. A forma de garantir que qualquer uma das avaliações pode ter múltiplos motivos associados, é adicionando a estrutura necessária para a construção de uma ligação muitos-para-muitos entre tabela de factos de cada uma das avaliações e a dimensão Motivo.

Para permitir a construção de uma ligação muitos-para-muitos é ainda necessário adicionar uma dimensão auxiliar, que contém apenas a chave que identifica a avaliação, e uma tabela ponte, onde é feito o mapeamento entre as avaliações e os seus respetivos motivos. A Figura 36 apresenta o modelo utilizado na construção do cubo de qualidade.

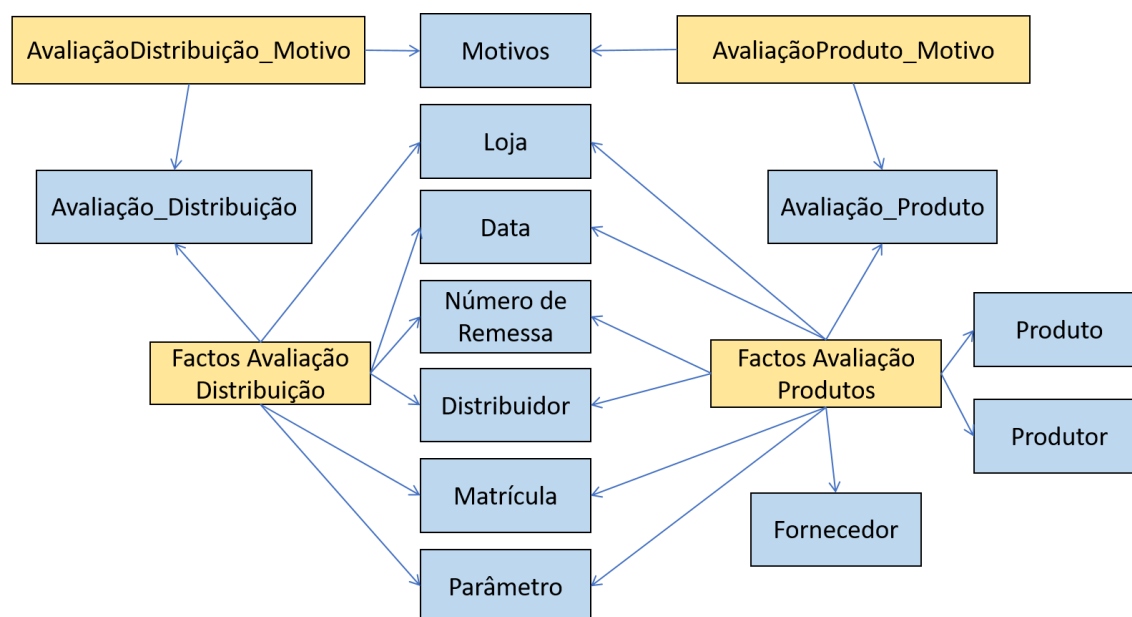


Figura 36 - Modelo utilizado na construção do cubo de qualidade

Com a estrutura do modelo concluída é então realizada a modelação das dimensões, tabelas de factos e tabelas pontes utilizadas neste modelo. À semelhança do descrito no modelo de indicadores, também neste modelo, para cada uma das dimensões e tabelas, foi definido o nome, uma breve descrição, o estado de visibilidade para cada coluna, sendo esta informação complementada com um exemplo do respetivo conteúdo. É de notar que a dimensão Loja e a dimensão Data não serão aqui descritas por já o terem sido no modelo de indicadores.

A dimensão Motivo contém a informação sobre quais os motivos possíveis que levam às não conformidades na avaliação da distribuição ou na avaliação dos produtos. Esta dimensão encontra-se visível para o utilizador quando acede ao cubo e a sua estrutura encontra-se representada na Tabela 12.

Nome da coluna	Descrição	Visível	Exemplo
PKReason	Chave primária	Não	1
BKReason	Chave de negócio que identifica o motivo	Não	2T529525-K65TC-29Y7-AF62-L9I86F8SFFS6
Código	Código do motivo	Sim	2
Descrição	Descrição do motivo	Sim	Temperatura do produto superior ao limite
Origem	Identifica se o motivo de não conformidade está relacionado	Sim	Qualidade

	com a qualidade do produto (Qualidade) ou se está relacionado com a distribuição (Compras)		
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 12 - Dimensão motivo

A dimensão Número de Remessa contém a informação do número do documento que identifica a que a entrega feita. Esta dimensão é degenerada, sendo que a sua informação provem das tabelas de factos. A estrutura desta dimensão está representada na Tabela 13.

Nome da coluna	Descrição	Visível	Exemplo
Numero Documento	Número do documento que identifica a entrega	Sim	2016-1-24738

Tabela 13 – Dimensão número de remessa

A dimensão Distribuidor, Fornecedor e Produtor apresentam exatamente a mesma estrutura, contendo cada uma delas a informação dos distribuidores, fornecedores e produtores, respetivamente. Estas três dimensões são visíveis ao utilizador quando acede ao cubo. A Tabela 14 apresenta a estrutura destas dimensões, sendo utilizada a dimensão Distribuidor como exemplo.

Nome da coluna	Descrição	Visível	Exemplo
PKDistributor	Chave primária	Não	1
BKDistributor	Chave de negócio que identifica o utilizador	Não	2T235525-K65TC-29Y7-BJY2-LS7YO98SFFS6
Nome	Nome do distribuidor	Sim	2
Morada	Morada da sede do distribuidor	Sim	M02
Telefone	Contacto telefónico do distribuidor	Sim	Temperatura do produto superior ao limite

Código postal	Código postal da sede	Sim	2645-340
Localidade	Localidade da sede	Sim	Lisboa
Email	Contacto de email do distribuidor	Sim	distribuicoes@distribuidor.com
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 14 - Dimensão distribuidor

A dimensão Produto contém informação de todos os produtos utilizados pelo cliente, estando a estrutura desta dimensão representada na Tabela 15.

Nome da coluna	Descrição	Visível	Exemplo
PKProduct	Chave primária	Não	1
BKProduct	Chave de negócio que identifica o produto	Não	2T529525-K65TC-29Y7-AF62-L9I86F8SFFS6
Descrição	Breve descrição do produto	Sim	Doce d'Avó
Unidade de medida	Unidade de medida utilizada	Sim	Unitário
Grupo	Grupo a que o produto pertence	Sim	Vendas
Família	Família de produto	Sim	Sobremesas
Subfamília	Subfamília do produto	Sim	Topping
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 15 – Dimensão produto

A dimensão Matrícula contém a informação do veículo que realizou a entrega do(s) produto(s). Esta é uma dimensão degenerada, ou seja, é construída com base na coluna correspondente presente na tabela de factos. Esta dimensão é visível para o utilizador e a sua estrutura é apresentada na Tabela 16.

Nome da coluna	Descrição	Visível	Exemplo
Matricula	Matrícula do veículo que realizou a entrega	Sim	12-AA-12

Tabela 16 - Dimensão matrícula

A dimensão Parâmetro contém a informação de todos os parâmetros utilizados na avaliação da distribuição e na avaliação do produto. Esta dimensão encontra-se visível ao utilizador e a sua estrutura está representada na Tabela 17.

Nome da coluna	Descrição	Visível	Exemplo
PKParameter	Chave primária	Não	1
BKParameter	Chave de negócio que identifica o utilizador	Não	2T529525-K65TC-29Y7-AF62-L9I86F8SFFS6
Código	Nome do utilizador	Sim	P03
Descrição	Cargo que o utilizador desempenha	Sim	Temperatura do produto congelado
Mínimo	Temperatura mínima a que o produto pode ter ou ser transportado	Sim	- 20°C
Máximo	Temperatura máxima a que o produto pode ter ou ser transportado	Sim	- 5°C
Observação	Observação que seja considerada relevante no ato da receção do produto	Sim	-
Valid From	Data em que o registo foi criado	Não	2015-01-01
Valid To	Data em que o registo ficou desatualizado	Não	9999-12-31
Is Active	Identifica se o registo está ativo	Não	1

Tabela 17 - Dimensão parâmetro

A dimensão Avaliação_Distribuição e a dimensão Avaliação_Produto possuem a mesma estrutura, sendo por isso apresentadas em conjunto e, cada uma delas, respetivamente, contém a chave que identifica cada avaliação. A estrutura destas tabelas

está representada na Tabela 18, sendo utilizada a Avaliação_Distribuição como exemplo. Estas duas dimensões não se encontram visíveis para o utilizador.

Nome da coluna	Descrição	Visível	Exemplo
PKFactAvDistribution	Chave primária	Não	1

Tabela 18 - Dimensão avaliação_distribuição

As tabelas ponte AvaliaçãoDistribuição_Motivo e AvaliaçãoProduto_Motivo também apresentam a mesma estrutura, e cada uma delas faz o mapeamento das suas avaliações com os motivos que estão na origem das suas não conformidades. Estas duas tabelas ponte não se encontram visíveis para o utilizador e a sua estrutura encontra-se representada na Tabela 19. À semelhança dos casos anteriores, o exemplo utilizado será o da tabela ponte AvaliaçãoDistribuição_Motivo.

Nome da coluna	Descrição	Visível	Exemplo
PKFactAvDistribution	Chave que identifica a avaliação	Não	1
PKReason	Chave que identifica o motivo da avaliação	Não	2

Tabela 19 - Tabela ponte AvaliaçãoDistribuição_Motivo

A tabela de Factos Avaliação Distribuição contém a informação de todas as avaliações às distribuições efetuadas, as chaves de ligação a todas as dimensões envolvidas na avaliação da distribuição e ainda algumas métricas. A estrutura desta tabela encontra-se na Tabela 20.

Nome da coluna	Descrição	Visível	Exemplo
PKFactAvDistribution	Chave primária	Não	1
FKStore	Chave estrangeira que identifica a loja	Não	3
FKDate	Chave estrangeira que identifica a data	Não	20150101
FKDistributor	Chave estrangeira que identifica o distribuidor	Não	5
FKParameter	Chave estrangeira que identifica o parâmetro	Não	2

Número de Remessa	Número do documento que identifica a entrega	Sim	2016-1-24738
Matricula	Matricula do veiculo que realizou a distribuição	Sim	12-AA-12
Número de conformidades	Número de distribuições que cumprem todos os parâmetros	Sim	1
Número de não conformidades	Número de distribuições que não cumprem todos os parâmetros	Sim	0
Número de devoluções	Número de distribuições que por não cumprirem os parâmetros levam a devoluções	Sim	0
Número de não verificações	Número de distribuições não verificadas	Sim	0
Temperatura	Temperatura registada na distribuição	Sim	-5

Tabela 20 - Tabela de Factos Avaliação Distribuição

A tabela de Factos Avaliação Produtos contém a informação de todas as avaliações realizadas aos produtos em específico após a receção dos mesmos nas lojas. Esta tabela de factos contém também as chaves de ligação a todas as dimensões deste modelo e ainda algumas métricas. A estrutura desta tabela é apresentada na Tabela 21.

Nome da coluna	Descrição	Visível	Exemplo
PKFactAvProduct	Chave primária	Não	1
FKStore	Chave estrangeira que identifica a loja	Não	3
FKDate	Chave estrangeira que identifica a data	Não	20150101
FKDistributor	Chave estrangeira que identifica o distribuidor	Não	5
FKParameter	Chave estrangeira que identifica o parâmetro	Não	2
FKProduct	Chave estrangeira que identifica o produto	Não	45

FKProducer	Chave estrangeira que identifica o produtor	Não	3
FKSupplier	Chave estrangeira que identifica o fornecedor	Não	20
Número de Remessa	Número do documento que identifica a entrega	Sim	2016-1-24738
Matricula	Matricula do veiculo que realizou a distribuição	Sim	12-AA-12
Número de conformidades	Número de distribuições que cumprem todos os parâmetros	Sim	1
Número de não conformidades	Número de distribuições que não cumprem todos os parâmetros	Sim	0
Número de devoluções	Número de distribuições que por não cumprirem os parâmetros levam a devoluções	Sim	0
Número de não verificações	Número de distribuições não verificadas	Sim	0
Temperatura	Temperatura registada na distribuição	Sim	-5
Quantidade Recebida	Quantidade de produto recebido	Sim	100
Quantidade Devolvida	Quantidade de produto devolvido	Sim	20

Tabela 21 - Tabela de Factos Avaliação Produtos

Com a estrutura do modelo e de todas as dimensões, tabelas ponte e tabelas de factos finalizadas, é então construído o cubo, utilizado para analisar o modelo de qualidade.

Porém, para que o cubo fique completo, é necessário ainda acrescentar as métricas calculadas identificadas no levantamento de requisitos, sendo estas:

- Valor médio da temperatura
- Percentagem de avaliações conformes
- Percentagem de avaliações não conformes
- Percentagem de avaliações não verificadas
- Percentagem de devoluções

Estas métricas existem para a avaliação da distribuição e para a avaliação dos produtos, sendo que foi utilizado, como exemplo, o código MDX criado para as métricas de avaliação da distribuição, visível na Figura 37.

```
CREATE MEMBER CURRENTCUBE.[Measures].[Valor médio de temperatura - Distribuição]
AS IIF(IsEmpty([Measures].[Temperature Value Count - Distribuição]) OR [Measures].[Temperature Value Count - Distribuição] = 0, null,
[Measures].[Temperatura - Distribuição] / [Measures].[Temperature Value Count - Distribuição]),
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'Avaliação da Distribuição';

CREATE MEMBER CURRENTCUBE.[Measures].[% Avaliações conformes - Distribuição]
AS [Measures].[Nº Conformidades - Distribuição] / [Measures].[Nº de Avaliações - Distribuição],
FORMAT_STRING = "Percent",
VISIBLE = 1 , DISPLAY_FOLDER = 'Porcentagens' , ASSOCIATED_MEASURE_GROUP = 'Avaliação da Distribuição';

CREATE MEMBER CURRENTCUBE.[Measures].[% Avaliações não conformes - Distribuição]
AS [Measures].[Nº não Conformidades - Distribuição] / [Measures].[Nº de Avaliações - Distribuição],
FORMAT_STRING = "Percent",
VISIBLE = 1 , DISPLAY_FOLDER = 'Porcentagens' , ASSOCIATED_MEASURE_GROUP = 'Avaliação da Distribuição';

CREATE MEMBER CURRENTCUBE.[Measures].[% Avaliações não verificadas - Distribuição]
AS [Measures].[Nº de não Verificações - Distribuição] / [Measures].[Nº de Avaliações - Distribuição],
FORMAT_STRING = "Percent",
VISIBLE = 1 , DISPLAY_FOLDER = 'Porcentagens' , ASSOCIATED_MEASURE_GROUP = 'Avaliação da Distribuição';

CREATE MEMBER CURRENTCUBE.[Measures].[% Devoluções - Distribuição]
AS [Measures].[Nº de Devoluções - Distribuição] / [Measures].[Nº de Avaliações - Distribuição],
FORMAT_STRING = "Percent",
VISIBLE = 1 , DISPLAY_FOLDER = 'Porcentagens' , ASSOCIATED_MEASURE_GROUP = 'Avaliação da Distribuição';
```

Figura 37 - Código MDX utilizado no cálculo das métricas referentes à avaliação da distribuição.

Com a implementação das métricas calculadas relativas à avaliação da distribuição e à avaliação dos produtos, o desenvolvimento do modelo de qualidade está terminado e pronto a ser consultado.

3.6 Processo de execução diário

Com a criação de todos os processos de ETL e a implementação de todas as estruturas na base de dados multidimensional, todos os componentes base da solução encontram-se desenvolvidos. Contudo, foi necessário criar um processo que junte todos estes componentes e os execute de forma ordenada e automática, para que diariamente sejam integrados os novos dados na solução.

Este processo de execução diário foi desenvolvido com o recurso a dois pacotes de SSIS, denominados de SSIS_Unipartner_MasterPackage (*MasterPackage*) e SSIS_Unipartner_MasterPackage_Execution (*MasterPackageExecution*) e a uma tabela de configuração na base de dados de configuração *PackageExecutionConfiguration*.

O *MasterPackage* é utilizado para construir toda a lógica de execução dos componentes. Na construção desta sequência de execução teve-se em conta todas as dependências dos processos de ETL e do carregamento da base de dados

multidimensional. Com o objetivo de ter o menor tempo de processamento possível foi, sempre que permitido, paralelizada a execução dos diversos processos. A estrutura final do *MasterPackage* (ver Apêndice A) está representada na Figura 38.

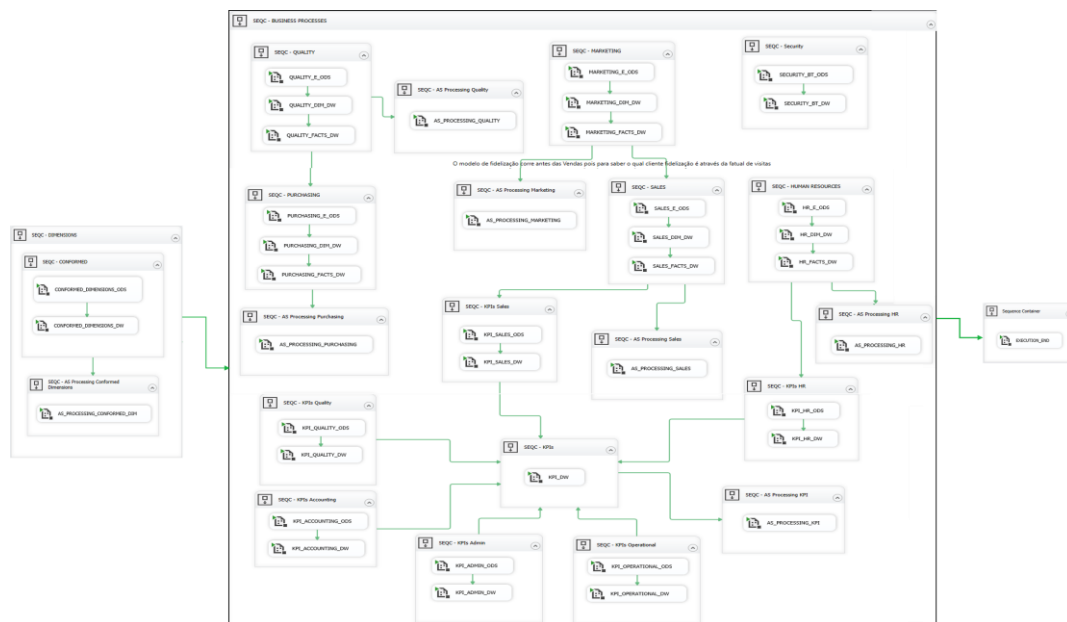


Figura 38 - Estrutura do MasterPackage da solução

À semelhança dos pacotes de ETL anteriormente descritos, este pacote é composto por diversos *containers* e, neste caso, possuem um ou mais *Execution Package Task* (EPT).

Os *containers* são utilizados para agrupar os *Execution Package Task* de acordo com a área de negócio que estes são responsáveis por executar.

Relativamente aos *Execution Package Task*, cada um deles tem um nome específico e todos eles identificam claramente qual o grupo de pacotes do processo de ETL que vão executar, como por exemplo *QUALITY_E_ODS*. No caso deste exemplo, o EPT correspondente é responsável por executar todos os pacotes de extração de dados referentes ao modelo de qualidade. Na verdade, o que os componentes EPT fazem é executar o pacote *MasterPackageExecution* passando-lhe como variável o seu próprio nome. O *MasterPackageExecution* é responsável por ir à tabela da base de dados de configuração e obter a lista de todos os pacotes do processo de ETL que estão incluídos nesse grupo e posteriormente executar cada um deles. A estrutura do pacote *MasterPackageExecution* encontra-se representada na Figura 39.

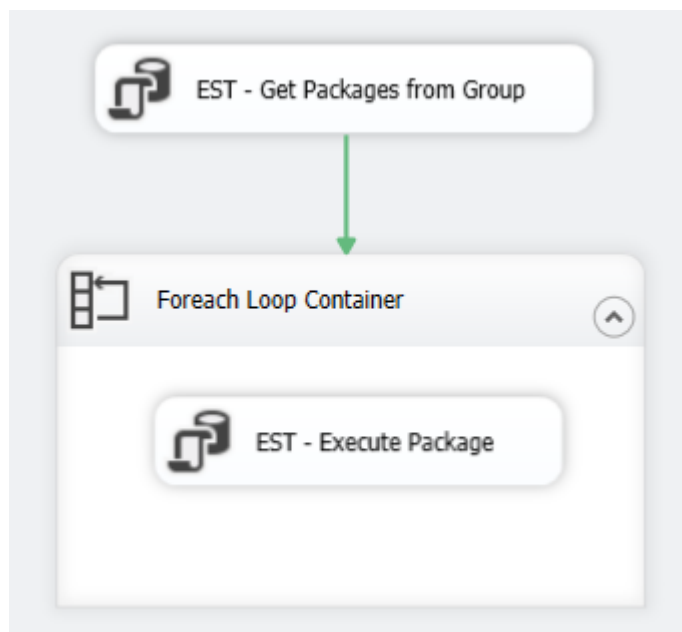


Figura 39 - Estrutura do MasterPackageExecution

Para concluir e para que todo o processo seja executado diariamente e de forma totalmente automática, foi criado um *job* no *SQL Server Agent Job* que é responsável pela execução do *MasterPackage*.

3.7 Camada de Visualização

A camada de visualização, que foi especificada conjuntamente com o cliente, consiste na última fase da solução de *Business Intelligence*. Esta camada, construída sobre a base de dados multidimensional, é a responsável por disponibilizar aos utilizadores a informação de forma visual, proporcionando um acesso simples à informação, permitindo uma interpretação rápida e objetiva dos indicadores necessários à tomada de decisão e acompanhamento da evolução do negócio tanto, na perspetiva global, como na perspetiva de cada marca ou na perspetiva individual de loja. Permite também uma navegação fácil, intuitiva e interativa.

Na construção da camada de visualização foi utilizado o Power BI para a criação dos *dashboards*.

Para auxiliar todo o processo de especificação de requisitos referentes à definição dos *dashboards* e facilitar as escolhas do cliente, ao mesmo tempo que lhe são demonstradas algumas das potencialidades da ferramenta na construção dos mesmos, foram desenvolvidos alguns *dashboards* exemplificativos.

A construção de *dashboards* para cada um dos modelos, foi realizada em conjunto com o cliente, com uma perspetiva formativa para que este venha a ter internamente a capacidade de desenvolver novos *dashboards* sobre os diversos modelos sempre que assim o pretender.

Os *dashboards* criados para demonstração ao cliente foram realizados sobre o modelo de indicadores e encontram-se representados nas figuras que são a seguir apresentadas.

No *dashboard* apresentado na Figura 40, é possível observar os diversos valores e *rankings* referentes aos indicadores de Vendas, Custo de Vendas e Margem Bruta referentes ao mês de maio. Conjuntamente com esta análise mensal é apresentado um gráfico de barras com os valores correspondentes aos vários meses do ano de 2016, mostrando a sua evolução, e permitindo a comparação do mês detalhado com os restantes meses do ano.



Figura 40 - Exemplo 1 de um dashboard

Na Figura 41 é apresentado outro exemplo de *dashboard*, que permite a análise de um conjunto mais alargado de indicadores, nomeadamente Vendas, Custo de Vendas, Gastos com Pessoal, Gastos Gerais e EBITDA. Para estes indicadores é possível verificar o seu *ranking* e analisar os seus valores, confrontando-os com os respetivos valores orçamentados para o mês de outubro, sendo esta análise realizada recorrendo a um tipo de representação gráfica diferente.

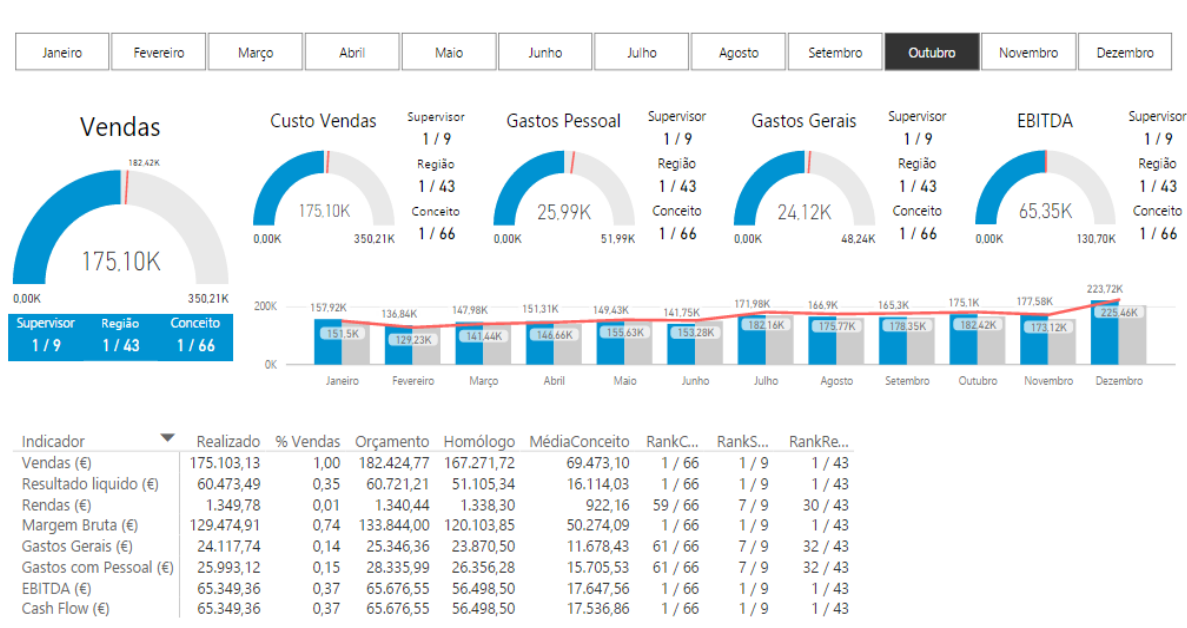


Figura 41 - Exemplo 2 de um dashboard

O dashboard presente na Figura 42 permite a análise de alguns dos indicadores para o mês de junho de outra forma distinta, fornecendo a informação detalhada de cada um deles com recurso a pequenas tabelas e apresentando a evolução do valor do respetivo indicador.

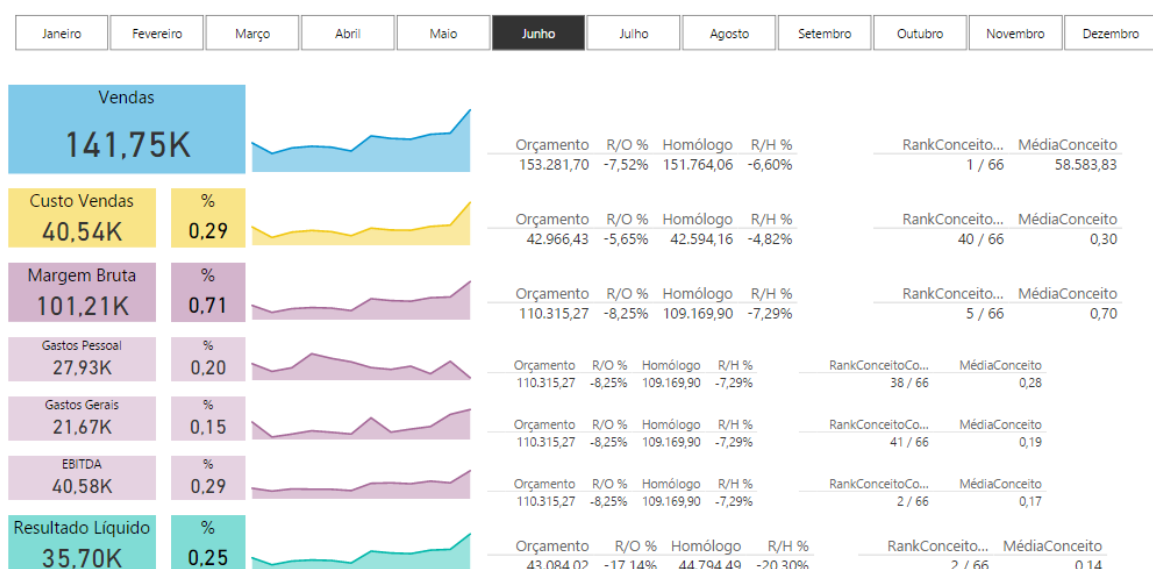


Figura 42 - Exemplo 3 de um dashboard

Capítulo 4 Conclusão e trabalho futuro

Neste capítulo é feita uma pequena análise de todo o trabalho desenvolvido incidindo sobre as principais contribuições que o projeto proporcionou ao cliente, quais as principais competências adquiridas e as principais dificuldades sentidas, terminando com algumas sugestões para o trabalho futuro.

4.1 Principais Contribuições

Esta solução de *Business Intelligence* teve como principal objetivo dotar a organização de uma ferramenta, que de uma forma expedita, permitisse obter de forma eficiente uma visão de toda a informação de negócio de uma forma consolidada e estruturada, para auxiliar nos processos de gestão e tomada de decisão respeitantes a todas as áreas do seu negócio.

Antes da solução estar implementada o cliente realizava todas as suas análises recorrendo a diversos documentos que se encontravam espalhados pelas diversas áreas da organização. Todos estes documentos tinham um foco de análise bastante específico e eram gerados de forma manual sempre que era necessário atualizar a informação. O crescimento exponencial da operação do cliente rapidamente levou a que este método fosse considerado insustentável, tendo sido solicitado à Unipartner a conceção e implementação da solução de BI.

Com a implementação da solução, o cliente passou a ter a informação de todas as suas áreas de negócio consolidadas e disponíveis a partir de um único ponto, com o acesso à informação a ser realizado através de uma interface única, garantindo também a capacidade deste realizar análises extremamente dinâmicas e flexíveis sobre todos os seus dados, podendo estas análises serem realizadas com o nível de complexidade que for pretendido. A solução garante ainda que os seus dados são atualizados diariamente de uma forma totalmente automática.

Em suma, com esta solução o cliente consegue obter uma grande visibilidade sobre os seus dados, permitindo-o retirar, em tempo útil, muito mais informação sobre o seu negócio, trabalhá-la e analisá-la, estando assim melhor capacitado para tomar decisões e ter mais tempo para se focar no seu negócio e na sua expansão.

Em todo este projeto, o meu contributo focou-se no levantamento de requisitos, análise funcional e construção dos modelos de qualidade e de indicadores, que foram anteriormente abordados neste documento. Para cada um destes dois modelos estive diretamente envolvido no desenho do modelo multidimensional implementado, criei os

processos de ETL, efetuei o desenvolvimento do modelo multidimensional do cubo, implementei o mecanismo de segurança utilizado no modelo dos indicadores e a criei as métricas calculadas sobre cada um destes modelos.

Adicionalmente, estive também envolvido ativamente na construção de todos os outros modelos e elementos que constituem a solução, tais como processos de ETL e estruturas necessárias para os mesmos, a criação e estruturação da base de dados multidimensional e, finalmente, a construção de *dashboards*.

4.2 Competências Adquiridas

No decorrer do PEI tive oportunidade de consolidar e aprofundar os conhecimentos sobre as soluções de BI, colocando-os em prática nos processos de levantamentos de requisitos, análise funcional, desenho de arquitetura e implementação da solução.

Esta exposição a diferentes fases do projeto permitiu também aprofundar os conhecimentos nas várias tecnologias utilizadas e desenvolver diferentes competências técnicas.

Na construção dos processos ETL que compõem a solução, são de destacar os conhecimentos obtidos relativamente à gestão de bases de dados e à utilização da ferramenta SSIS para a construção de diversos fluxos.

Na construção dos modelos multidimensionais utilizados, foi possível consolidar alguns dos conhecimentos prévios e ganhar conhecimentos com a resolução dos diversos desafios que os modelos de indicadores e de qualidade colocaram. Nesta etapa destacam-se os novos conhecimentos referentes à ferramenta SSAS e a aprendizagem de MDX.

Na última etapa da solução, a criação de relatórios para demonstração ao cliente, foi explorada a ferramenta Power BI e as suas potencialidades.

Este projeto, por ter sido desenvolvido em equipa e por me ter proporcionado algum contacto com o cliente, permitiu o desenvolvimento de várias *soft skills*, como por exemplo, análise crítica e gestão de conflitos.

4.3 Principais Dificuldades

Na realização deste projeto destacam-se dois desafios que foram ultrapassados com sucesso, mas que tiveram algum impacto no tempo de implementação da solução.

O principal desafio, que foi o que mais impactou o tempo de realização do projeto, foi o levantamento de requisitos junto do cliente. Após o levantamento de requisitos

iniciais, e por se optar por uma metodologia de desenvolvimento ágil, por vezes ocorriam algumas alterações aos requisitos, causando estas muitas vezes um impacto significativo no planeamento feito até então.

A criação de *dashboards* finais para o cliente foi também um tópico que colocou alguns desafios. A criação dos *dashboards* exemplo para as diversas áreas implica um entendimento bastante detalhado da área e do modelo de negócio associado.

Para além disso, como o cliente não possui uma ideia clara sobre os relatórios que pretende obter e qual a informação que neles deve constar, o desafio consiste em construir mais e mais relatórios do que aqueles que tinham sido acordados. Face a esta situação, decidiu-se elaborar algumas formações para que o próprio cliente tenha a capacidade e autonomia para construir alguns dos relatórios complementares que pretende.

4.4 Trabalho Futuro

Estando a solução acordada com cliente concluída, a fase seguinte passa pela implementação de otimizações que possam vir a ser necessárias e pela prestação de serviços de suporte e manutenção.

Os objetivos de evolução da solução passam pela implementação de novos modelos que o cliente possa vir a ter necessidade de desenvolver devido à constante evolução do seu negócio e posicionamento nos vários mercados onde está presente, ao dinamismo que imprime à sua oferta e aos processos inerentes ao funcionamento das suas lojas.

Por ser uma ferramenta recente, o Power BI, tem sofrido uma evolução bastante acentuada em todas as suas funcionalidades, sendo por isso necessário ir complementando junto do cliente a formação efetuada para a criação de relatórios. Esta tem como objetivo garantir sempre a melhor experiência possível para o utilizador na geração dos *dashboards* e na análise da informação.

Bibliografia

1. Power BI: <https://powerbi.microsoft.com/en-us/> (acedido a 21 de dezembro de 2016)
2. Power BI Row-level security: <https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-rls/> (acedido a 21 de dezembro de 2016)
3. Microsoft SQL Server Database: <https://www.microsoft.com/en-us/sql-server/sql-server-2016> (acedido a 18 de dezembro de 2016)
4. SQL Server Database Engine: [https://technet.microsoft.com/en-us/library/ms187875\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms187875(v=sql.105).aspx) (acedido a 18 de dezembro de 2016)
5. Microsoft SQL Server Integration Services: <https://msdn.microsoft.com/pt-pt/library/ms141026.aspx> (acedido a 18 de dezembro de 2016)
6. Integration Services (SSIS) Packages: <https://msdn.microsoft.com/en-us/library/ms141134.aspx> (acedido a 18 de dezembro de 2016)
7. Microsoft SQL Server Analysis Services: [https://technet.microsoft.com/en-us/library/ms175609\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/ms175609(v=sql.90).aspx) (acedido a 18 de dezembro de 2016)
8. Analysis Services: <https://msdn.microsoft.com/en-us/library/bb522607.aspx> (acedido a 18 de dezembro de 2016)
9. Matriz de processos: <http://www.kimballgroup.com/1999/12/the-matrix/> (acedido a 19 de dezembro de 2016)
10. Row-Level Security: <https://msdn.microsoft.com/en-us/library/dn765131.aspx> (acedido a 21 de dezembro de 2016)
11. Ralph Kimball e Margy Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, Wiley, 3ª edição, 2013
12. Mitch Tulloch with the Windows Azure Team, *Introducing Windows Azure For IT Professionals*
13. Microsoft Azure: <https://azure.microsoft.com/pt-pt/> (acedido a 19 de dezembro de 2016)
14. Carlos Pampumlim Caldeira, *Data Warehousing Conceitos e modelos*, 2ª edição, Edições Sílabo, 2012
15. Maribel Yasmina Santos e Isabel Ramos, *Business Intelligence Tecnologias da Informação na Gestão de Conhecimento*, 2ª edição, FCA – Editora de Informática, setembro 2015,

16. Metodologia de trabalho Ágil: <http://scrummethodology.com/> (acedido a 26 de junho de 2017)
17. Roles no Analysis Services: <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/roles-and-permissions-analysis-services> (acedido a 10 de Agosto de 2017)
18. *Role-playing*: <http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/role-playing-dimension/> (acedido a 19/06/2017)

Apêndice A

4.5 Master Package

